NASA/CR-2000-210302
ICASE Report No. 2000-28

The Intuitionism Behind Statecharts Steps

Gerald Luettgen
ICASE, Hampton, Virginia

Michael Mendler
The University of Sheffield, Sheffield, England

July 2000

The NASA STI Program Office . . . in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA Scientific and Technical Information (STI) Program Office plays a key part in helping NASA maintain this important role.

The NASA STI Program Office is operated by Langley Research Center, the lead center for NASA's scientific and technical information. The NASA STI Program Office provides access to the NASA STI Database, the largest collection of aeronautical and space science STI in the world. The Program Office is also NASA's institutional mechanism for disseminating the results of its research and development activities. These results are published by NASA in the NASA STI Report Series, which includes the following report types:

- TECHNICAL PUBLICATION. Reports of completed research or a major significant phase of research that present the results of NASA programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA counter-part or peer-reviewed formal professional papers, but having less stringent limitations on manuscript length and extent of graphic presentations.

- TECHNICAL MEMORANDUM. Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.

- CONTRACTOR REPORT. Scientific and technical findings by NASA-sponsored contractors and grantees.

- CONFERENCE PUBLICATIONS. Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or co-sponsored by NASA.

- SPECIAL PUBLICATION. Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.

- TECHNICAL TRANSLATION. English-language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services that help round out the STI Program Office's diverse offerings include creating custom thesauri, building customized databases, organizing and publishing research results . . . even providing videos.

For more information about the NASA STI Program Office, you can:

- Access the NASA STI Program Home Page at http://www.sti.nasa.gov/STI-homepage.html

- Email your question via the Internet to help@sti.nasa.gov

- Fax your question to the NASA Access Help Desk at (301) 621-0134

- Phone the NASA Access Help Desk at (301) 621-0390

- Write to:
NASA Access Help Desk
NASA Center for AeroSpace Information
7121 Standard Drive
Hanover, MD 21076-1320

NASA/CR-2000-210302
ICASE Report No. 2000-28

The Intuitionism Behind Statecharts Steps

Gerald Luettgen
ICASE, Hampton, Virginia

Michael Mendler
The University of Sheffield, Sheffield, England

Institute for Computer Applications in Science and Engineering
NASA Langley Research Center
Hampton, VA

Operated by Universities Space Research Association

National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23681-2199

Prepared for Langley Research Center
under Contract NAS1-97046

July 2000

Available from the following:

NASA Center for AeroSpace Information (CASI)
7121 Standard Drive
Hanover, MD 21076-1320
(301) 621-0390

National Technical Information Service (NTIS)
5285 Port Royal Road
Springfield, VA 22161-2171
(703) 487-4650

THE INTUITIONISM BEHIND STATECHARTS STEPS*

GERALD LÜTTGEN[†] AND MICHAEL MENDLER[‡]

Abstract. The semantics of Statecharts macro steps, as introduced by Pnueli and Shalev, lacks compositionality. This report first analyzes the compositionality problem and traces it back to the invalidity of the Law of the Excluded Middle. It then characterizes the semantics via a particular class of linear, intuitionistic Kripke models, namely stabilization sequences. This yields, for the first time in the literature, a simple fully–abstract semantics which interprets Pnueli and Shalev's concept of failure naturally. The results not only give insight into the semantic subtleties of Statecharts, but also provide a basis for an implementation, for developing algebraic theories for macro steps, and for comparing different Statecharts variants.

Key words. Statecharts, compositionality, full abstraction, intuitionistic Kripke semantics

Subject classification. Computer Science

1. Introduction. *Statecharts* is a well–known visual design notation for specifying the behavior of *reactive systems* [7]. It extends *finite state machines* with concepts of (i) *hierarchy*, so that one may speak of a state as having sub–states, (ii) *concurrency*, thereby allowing the definition of systems having simultaneously active sub–systems, and (iii) *priority*, such that one may express that certain system activities have precedence over others. The success of Statecharts in the software–engineering community is founded on the language's capability for intuitively modeling the complex control aspects inherent in many software systems. However, the search for a practically and theoretically satisfying semantics for Statecharts is still actively pursued at many academic and industrial research laboratories and has led to the definition of numerous Statecharts variants [20].

In a seminal paper, Pnueli and Shalev presented two equivalent formalizations of Statecharts semantics [17]. According to their semantic model, a Statechart may *respond* to an event entering the system by engaging in an enabled transition. This may generate new events which, by *causality*, may in turn trigger additional transitions while disabling others. The *synchrony hypothesis* ensures that one execution step, a so–called *macro step*, is complete as soon as this chain reaction comes to a halt. Unfortunately, Pnueli and Shalev's semantics violates the desired property of *compositionality* which is a prerequisite for modular analyses of Statecharts specifications as well as for compositional code generation. Huizing and Gerth [10] showed that combining compositionality, causality, and the synchrony hypothesis cannot be done within a simple, single–leveled semantics. Some researchers then devoted their attention to investigating new variants of Statecharts, obeying just two of the three properties. In ESTEREL [3] and ARGOS [16], causality is treated separately from compositionality and synchrony, while in (synchronous) STATEMATE [8] the synchrony hy-

[*] This research was supported by the National Aeronautics and Space Administration under NASA Contract No. NAS1–97046 while the authors were in residence at the Institute for Computer Applications in Science and Engineering (ICASE), Mail Stop 132C, NASA Langley Research Center, Hampton, Virginia 23681–2199, USA.

[†] Institute for Computer Applications in Science and Engineering (ICASE), Mail Stop 132C, NASA Langley Research Center, Hampton, Virginia 23681–2199, USA, e–mail: luettgen@icase.edu.

[‡] Department of Computer Science, The University of Sheffield, 211 Portobello Street, Sheffield S1 5DP, England, e–mail: M.Mendler@dcs.shef.ac.uk.

pothesis is rejected. Other researchers achieved combining all three properties by storing complex semantic information via preorders [13, 15, 18] or transition systems [6, 14]. However, no analysis of exactly how much information is needed to achieve compositionality has been made so far.

This report first illustrates the compositionality defect of Pnueli and Shalev's semantics by showing that equality of response behavior is not preserved by the concurrency and hierarchy operators of Statecharts. The reason is that macro steps abstract from causal interactions with a system's environment, thereby imposing a closed–world assumption. Indeed, the studied problem can be further traced back to the invalidity of the *Law of the Excluded Middle.* To overcome the problem, we interpret Statecharts, relative to a given system state, as intuitionistic formulas. These are given meaning as specific *intuitionistic Kripke structures* [19], namely linear increasing sequences of event sets, called *stabilization sequences*, which encode interactions between Statecharts and environments. In this domain, which we characterize via semi–lattices and in which Pnueli and Shalev's semantics may be explained by considering a distinguished sub–domain, we develop a *fully–abstract* macro–step semantics in two steps. First, we study Statecharts without hierarchy operators which are in fact choice operators in our setting since we observe single macro steps only. We show that in this fragment, stabilization sequences naturally characterize the largest congruence contained in equality of response behavior. In the second step, based on a non–standard *distributivity* and *expansion law*, as well as our lattice–theoretic characterization of the intuitionistic semantics, we lift our results to arbitrary Statecharts. It is worth remarking that these results are achieved in a slightly extended Statecharts algebra that allows for general choice operators and also introduces explicit failure events. We show that this extension is conservative over the standard "visual" syntax of Statecharts. As a byproduct, this report suggests a natural way of admitting disjunctions in transition triggers, thereby solving a logical inadequacy of Pnueli and Shalev's setting. Moreover, our results build a foundation for an efficient implementation of Pnueli and Shalev's semantics that avoids backtracking, for algebraic characterizations of macro–step semantics, and also for comparisons among related Statecharts variants.

The remainder of this report is organized as follows. The next section presents our notation for Statecharts, recalls the classic Statecharts semantics of Pnueli and Shalev, and analyzes the compositionality problem. Sec. 3 presents a new intuitionistic semantics for Statecharts macro steps, characterizes Pnueli and Shalev's semantics within the novel framework, and provides a full-abstraction result for the Statecharts language without hierarchy operator. The latter result is extended to the full language in Sec. 4. Finally, Secs. 5 and 6 discuss related work and present our conclusions and directions for future work, respectively. The appendices contain some longer proofs as well as some complimentary technical material.

2. Statecharts: Syntax, Semantics, and Compositionality. Statecharts is a visual language for specifying *reactive systems*, i.e., concurrent systems interacting with their *environment.* They subsume labeled transition systems where labels are pairs of *event* sets. The first component of a pair is referred to as *trigger*, which may include *negative events*, and the second as *action.* Intuitively, a transition is enabled if the environment offers all events in the trigger but not the negative ones. When a transition fires, it produces the events specified in its action. Concurrency is introduced by allowing Statecharts to run in parallel and to communicate by *broadcasting* events. Additionally, *basic states* may be hierarchically refined by injecting other Statecharts. This creates composite states of two possible sorts, which are referred to as *and*–states and *or*–states, respectively. Whereas and–states permit parallel decompositions of states, or–states allow for sequential decompositions. Consequently, an and–state is *active* if all of its sub–states are active, and an or–state is active if exactly one of its sub–states is.

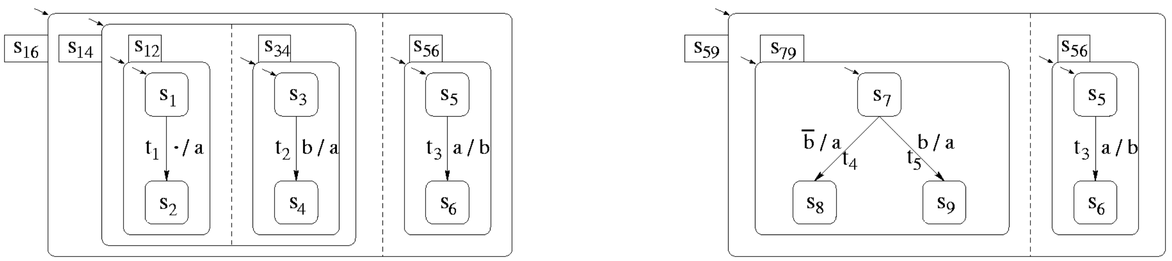

FIG. 2.1. *Two example Statecharts*

As an example, the Statechart in Fig. 2.1 on the left consists of and–state s_{16} which puts and–state s_{14} and or–state s_{56} in parallel. Similarly, state s_{14} is a parallel composition of or–states s_{12} and s_{34}. Each of these or–states describes a sequential state machine and is refined by two basic states. In case of s_{12}, basic state s_1 is the initial state which is connected to basic state s_2 via transition t_1. Here, s_1 is the source state of t_1, state s_2 is its target state, "·" symbolizes its empty trigger, and a is its action. Hence, t_1 is always enabled regardless of the events offered by the environment. Its firing produces event a and switches the active state of s_{12} from s_1 to s_2. This initiates a causal chain reaction, since the generation of a in turn triggers transition t_3 in parallel component s_{56} which introduces event b. As a consequence, transition t_2 in or–state s_{34} becomes enabled and fires within the same *macro step*.

The Statechart depicted in Fig. 2.1 on the right is like the one on the left, except that and–state s_{14} is replaced by or–state s_{79}. The latter state encodes a choice regarding the execution of transitions t_4 and t_5 from state s_7. The trigger of t_4 is $\overline{b}$, i.e., t_4 is triggered by the absence of event b. Starting with an environment offering no event, thus assuming b to be absent, and–state s_{59} can autonomously engage in t_4. The generation of a in turn triggers transition t_3 which fires and produces b. However, t_4 was fired under the assumption that b is absent. Since Statecharts is a synchronous language and no event can be simultaneously present and absent within the same macro step, this behavior is rejected as *globally inconsistent*. Thus, the response of s_{59} to the empty environment is failure, which is operationally different from an empty response.

2.1. Statecharts Configurations and Step Semantics. Like [17] we present the semantics of Statecharts as a single–step semantics which is given relative to a fixed but arbitrary set of active states. As a consequence, Statecharts' hierarchy operator acts exactly like a choice operator. Formally, let Π and $\mathcal{T}$ be countably infinite sets of events and transition names, respectively. For every event $e \in \Pi$, its negative counterpart is denoted by $\overline{e}$. We define $\overline{\overline{e}} =_{\mathrm{df}} e$ and write $\overline{E}$ for $\{\overline{e} \,|\, e \in E\}$. With every $t \in \mathcal{T}$, we associate a transition E/A consisting of a trigger $\mathsf{trg}(t) =_{\mathrm{df}} E \subseteq_{\mathrm{fin}} \Pi \cup \overline{\Pi}$ and an action $\mathsf{act}(t) =_{\mathrm{df}} A \subseteq_{\mathrm{fin}} \Pi$, where E and A are required to be finite sets. For simplicity, we use the abbreviation $e_1 \cdots e_n/a_1 \cdots a_m$ for transition $\{e_1, \ldots, e_n\}/\{a_1, \ldots, a_m\}$, and we denote an empty trigger or action in a transition by symbol '·'. We also write $P, \overline{N}/A$ for label E/A when we wish to distinguish the set $P =_{\mathrm{df}} E \cap \Pi$ of *positive* trigger events from the set $N =_{\mathrm{df}} \overline{E \cap \overline{\Pi}}$ of *negative* trigger events. Now, we are able to describe a Statechart relative to a set of active states as a term in the BNF

$$C \quad ::= \quad \mathbf{0} \quad | \quad x \quad | \quad t \quad | \quad C \,||\, C \quad | \quad C + C\,,$$

where $t \in \mathcal{T}$ and x is a variable. Terms not containing variables are called *configurations*. Intuitively, configuration $\mathbf{0}$ represents a Statechart state with no outgoing transitions (basic state), $C \,||\, D$ denotes the parallel composition of configurations C and D (and–state), and $C + D$ stands for the choice between executing C or D (or–state). As mentioned earlier, the latter construct $+$ coincides with Statecharts' hierarchy operator, which reduces to choice when considering single macro steps only; thus, we refer to operator $+$ also as choice operator. Moreover, in the visual Statecharts notation, $C + D$ is somewhat more restrictive, in that it requires D to be a choice of transitions. For instance, $(t_1 \parallel t_2) + (t_3 \parallel t_4)$ is prohibited in Statecharts visual syntax whereas it is a valid configuration in our setting. Semantically, however, our generalization is inessential with respect to the considered semantics of Pnueli and Shalev, as we will show in Sec. 4.4. The set of all configurations is denoted by C and ranged over by C and D. The set of "+"–free, or *parallel*, configurations is written as PC. We call terms $\Phi[x]$ with a single variable occurrence x *contexts*, and write $\Phi[C]$ for the substitution of C for x in $\Phi[x]$. Contexts of form $x \parallel C$ and $x + C$ are referred to as *parallel contexts* and *choice contexts*, respectively. We tacitly assume that transition names are unique in every term, and we let $\mathsf{trans}(C)$ stand for the set of transition names occurring in C.

Any Statechart in a given set of active states corresponds to a configuration. For example, Statecharts s_{14} and s_{79}, in their initial state, correspond to configurations $C_{14} =_{\mathrm{df}} t_1 \parallel t_2$ and $C_{79} =_{\mathrm{df}} t_4 + t_5$, respectively. The Statecharts depicted in Fig. 2.1 are then formalized as $C_{16} =_{\mathrm{df}} \Phi_{56}[C_{14}]$ and $C_{59} =_{\mathrm{df}} \Phi_{56}[C_{79}]$, respectively, using the parallel context $\Phi_{56}[x] =_{\mathrm{df}} x \parallel t_3$. Moreover, since transitions are uniquely named in configurations and thus may be associated with their source and target states, one can easily determine the set of active states reached after firing a set of transitions; see [17] for details. As in [17], we do not consider *interlevel transitions* and *state references* [20] to keep our syntax for Statecharts sufficiently simple. Although the syntax would have to be extended, our semantics can accommodate these features, too. Finally, we want to remark that the unique naming of transitions is not an essential assumption but just a convenient means in the operational semantics to define the step response of a Statechart configuration. We will see that the intuitionistic model theory developed in this report allows us to do away with naming transitions.

To present the *response behavior* of a configuration C, as defined by Pnueli and Shalev, we have to determine which transitions in $\mathsf{trans}(C)$ may fire together to form a macro step. A macro step comprises a *maximal* set of transitions that are *triggered* by events offered by the environment or produced by the firing of other transitions, that are mutually *consistent* ("orthogonal"), and that obey *causality* and *global consistency*. We start off by formally introducing some of these notions.

- A transition t is *consistent* with $T \subseteq \mathsf{trans}(C)$, in signs $t \in \mathsf{consistent}(C, T)$, if t is not in the same parallel component as any $t' \in T$. Formally,

$$\mathsf{consistent}(C, T) =_{\mathrm{df}} \{t \in \mathsf{trans}(C) \mid \forall t' \in T.\ t \triangle_C t'\},$$

  where $t \triangle_C t'$, if $t = t'$ or if t and t' are on different sides of an occurrence of $\parallel$ in C.
- A transition t is *triggered* by a finite set E of events, in signs $t \in \mathsf{triggered}(C, E)$, if the positive, but not the negative trigger events of t are in E. Formally,

$$\mathsf{triggered}(C, E) =_{\mathrm{df}} \{t \in \mathsf{trans}(C) \mid \mathsf{trg}(t) \cap \Pi \subseteq E \text{ and } \overline{(\mathsf{trg}(t) \cap \overline{\Pi})} \cap E = \emptyset\}.$$

- A transition t is *enabled* in C with respect to a finite set E of events and a set T of transitions, if $t \in \mathsf{enabled}(C, E, T)$ where

$$\mathsf{enabled}(C, E, T) =_{\mathrm{df}} \mathsf{consistent}(C, T) \cap \mathsf{triggered}(C, E \cup \bigcup_{t \in T} \mathsf{act}(t)).$$

  Intuitively, assuming transitions T are known to fire, $\mathsf{enabled}(C, E, T)$ determines the set of all transitions of C that are enabled by the actions of T and the environment events in E. In the following, we use $\mathsf{act}(T)$ as an abbreviation for $\bigcup_{t \in T} \mathsf{act}(t)$.

With these preliminaries, we may now present Pnueli and Shalev's iterative *step–construction procedure* [17] for causally determining macro steps relative to a configuration C and a finite set E of environment events.

```
procedure step–construction($C$, $E$);
  var $T := \emptyset$;
  while $T \subset \mathsf{enabled}(C, E, T)$ do
    choose $t \in \mathsf{enabled}(C, E, T) \setminus T$;
    $T := T \cup \{t\}$
  od;
  if $T = \mathsf{enabled}(C, E, T)$ then return $T$
  else report failure
end step–construction.
```

This procedure computes *nondeterministically*, relative to a configuration C and a finite environment E, those sets T of transitions that can fire together in a macro step. Note that due to failures raised when detecting global inconsistencies, the step construction might involve *backtracking*, which makes the above algorithm inefficient for implementation. To highlight the role of failures further in this report, it will be useful to introduce a special failure event $\bot \in \Pi$ in order to represent the failure behavior of the step semantics explicitly. For instance, we can then define transition $a/\bot$ which raises a failure exception as soon as event a becomes present. Note that, e.g., the firing of transition $\overline{a}/a$, which can already be expressed in the standard syntax, raises a failure in the absence of event a. Hence, adding an explicit $\bot$ event makes the representation of failure behavior more symmetric in that it allows us to enforce the *presence* as well as the *absence* of certain events in a macro step. It should be stressed that, as we will show in Sec. 4.4, adding event $\bot$ is a conservative extension that does not change the semantics of the original Statecharts language. It permits, however, a more uniform algebra of configurations. In particular, having $\bot$ available has the important technical advantage that certain semantic constructions on the original Statecharts language become syntactically representable. Moreover, there are also new behaviors expressible that may be useful in applications. Therefore, we will study both variants of Statecharts semantics, with and without $\bot$, in the remainder of this report.

Following Pnueli and Shalev, a set T of transitions is called *constructible*, for a given configuration C and a finite set E of environment events, if and only if it can be obtained as a result of successfully executing procedure *step–construction*. Whenever we wish to indicate the environment, we say that T is *E–constructible*. For each E–constructible set T, set $A =_{\mathrm{df}} E \cup \mathsf{act}(T) \subseteq_{\mathsf{fin}} \Pi$ is called the *(step) response* of C for E, in signs $C \Downarrow_E A$. If event $\bot$ is considered, we also require $\bot \notin A$. Moreover, if $E = \emptyset$, we simply write $C \Downarrow A$. Note that E may also be modeled by a parallel context consisting of a single transition $\cdot/E$, as $C \Downarrow_E A$ if and only if $(C \parallel \cdot/E) \Downarrow A$ holds. Pnueli and Shalev also provided an equivalent declarative definition of their operational step semantics. A set T of transitions is called *E–separable* for C if there exists a proper subset $T' \subset T$ such that $\mathsf{enabled}(C, E, T') \cap (T \setminus T') = \emptyset$. Further, T is *E–admissible* for C if (i) T is E–inseparable for C, (ii) $T = \mathsf{enabled}(C, E, T)$, and (iii) $\bot \notin \mathsf{act}(T)$. When configuration C and environment E are understood, we also say that T is *admissible* or *separable*, respectively.

THEOREM 2.1 (Pnueli & Shalev [17]). *For all configurations $C \in \mathsf{C}$ and event sets $E \subseteq_{\mathit{fin}} \Pi$, a set T of transitions is E–admissible for C if and only if T is E–constructible for C.*

While this theorem emphasizes the mathematical elegance of Pnueli and Shalev's semantics, it still does not support implementations. However, because of Thm. 2.1, one may confuse the notions of constructibility and admissibility. In fact, the approach we are going to present in the following sections is derived more conveniently from the declarative characterization.

2.2. The Compositionality Problem. The macro–step semantics induces a natural equivalence relation $\sim$ over configurations, called *step equivalence*, satisfying $C \sim D$, whenever $C \Downarrow_E A$ if and only if $D \Downarrow_E A$, for all $E, A \subseteq_{\mathrm{fin}} \Pi$. For simplicity, $\sim$ does not account for target states of transitions since these can be encoded in event names. The compositionality defect of the macro–step semantics manifests itself in the fact that $\sim$ is not a congruence for the configuration algebra. Consider our example of Fig. 2.1 and assume that states s_2, s_4, s_6, s_8, and s_9 are all equivalent. It is easy to see that configurations C_{14} and C_{79} have the same response behavior. Both $C_{14} \Downarrow_E A$ and $C_{79} \Downarrow_E A$ are equivalent to $A = E \cup \{a\}$, no matter whether event b is present or absent in environment E. However, $\Phi_{56}[C_{14}] = C_{16} \not\sim C_{59} = \Phi_{56}[C_{79}]$, since $C_{16} \Downarrow \{a, b\}$ but $C_{59} \not\Downarrow A$, for any $A \subseteq_{\mathrm{fin}} \Pi$, as C_{59} always fails for the empty environment. Hence, the equivalence $C_{14} \sim C_{79}$ is not preserved by context $\Phi_{56}[x]$. The intuitive reason for why C_{14} and C_{79} are identified in the first place is that the response semantics does not account for any proper interaction with the environment. It adopts the classic *closed–world assumption* which states that every event is either present from the very beginning of a given macro step or will never arise. This eliminates the possibility that events may be generated due to interactions with the environment, such as event b in $C_{16} \Downarrow \{a, b\}$. In short, a *compositional* macro–step semantics does not validate the *Law of the Excluded Middle* $b \vee \neg b = \mathit{true}$. Since *intuitionistic logic* [19] differs from classic logic by refuting the Law of the Excluded Middle, it is a good candidate framework for analyzing the step semantics of Statecharts.

It must be stressed that the compositionality defect is an issue of parallel composition $\|$ and not of operator $+$. Configuration $C_{79} = \overline{b}/a + b/a$ has exactly the same behavior as configuration $C'_{79} =_{\mathrm{df}} \overline{b}/a \parallel b/a$ which we could have used instead. The compositionality problem can also be demonstrated by the two parallel configurations $D_1 =_{\mathrm{df}} \cdot/a \parallel b/c$ and $D_2 =_{\mathrm{df}} \overline{b}/a \parallel b/ac$ which have the same step responses but can be distinguished in context $\Phi_{56}[x]$, as $\Phi_{56}[D_1] \Downarrow \{a, b, c\}$ but $\Phi_{56}[D_2] \not\Downarrow A$, for any $A \subseteq_{\mathrm{fin}} \Pi$.

Our goal is to characterize the largest congruence $\simeq$, called *step congruence*, contained in step equivalence, where $C \simeq D$, if $\Phi[C] \sim \Phi[D]$, for all contexts $\Phi[x]$. While the compositionality defect is well–known, a fully–abstract semantics with respect to Pnueli and Shalev's macro–step semantics has not yet been presented in the literature. Of course, one can trivially obtain that $C \simeq D$ is equivalent to $\llbracket C \rrbracket_0 = \llbracket D \rrbracket_0$, where $\llbracket C \rrbracket_0 =_{\mathrm{df}} \{\langle A, \Phi[x] \rangle \mid \Phi[C] \Downarrow A\}$. However, $\llbracket \cdot \rrbracket_0$ is a syntactic characterization rather than a semantic one, which we will develop below. Note that we intend to achieve compositionality in the declarative sense of a fully–abstract semantics and not in the constructive sense of a denotational semantics (cf. Sec. 5).

3. Intuitionistic Semantics via Stabilization Sequences. We start off by investigating parallel configurations within parallel contexts, for which many semantic insights may already be obtained. First, we propose a novel intuitionistic semantics for this fragment, then show its relation to Pnueli and Shalev's original semantics, and finally derive a full–abstraction result. The next section generalizes this result to arbitrary configurations within arbitrary contexts.

Our new semantic interpretation of parallel configurations C, based on an "open–world assumption," is given in terms of finite increasing sequences of *worlds* (or *states*) $E_0 \subset E_1 \subset \cdots \subset E_n$, for some natural number n. Each $E_i \subseteq_{\mathrm{fin}} \Pi \setminus \{\bot\}$ is the set of events generated or present in the respective world, and the absence of $\bot$ ensures that each world is consistent. A sequence represents the interactions between C and a potential environment during a macro step. Intuitively, the initial world E_0 contains all events e which are generated by those transitions of C that can fire autonomously. When transitioning from world E_{i-1} to E_i, some events in $E_i \setminus E_{i-1}$ are provided by the environment, as reaction to the events validated by C when reaching E_{i-1}. The new events destabilize world E_{i-1} and may enable a chain reaction of transitions within C. The step–construction procedure, which tracks and accumulates all these events, then defines the new world E_i. In accordance with this intuition, we call the above sequences *stabilization sequences.* The overall response of C after n interactions with the environment is the event set E_n.

The monotonicity requirement of stabilization sequences reflects the fact that our knowledge of the presence and absence of events increases in the process of constructing a macro step. More precisely, each world contains the events assumed or known to be present. Only if an event is not included in the final world, it is known to be absent for sure. The fact that an event e is not present in a world, $e \notin E(i)$, does not preclude e from becoming available later in the considered stabilization sequence. This semantic gap between "not present" and "absent" makes the underlying logic *intuitionistic* as opposed to classic. Indeed, we shall see that parallel configurations are most naturally viewed as intuitionistic formulas specifying linear intuitionistic Kripke models.

3.1. Intuitionistic Semantics for Parallel Configurations. Formally, a stabilization sequence M is a pair (n, V), where $n \in \mathbb{N} \setminus \{0\}$ is the *length* of the sequence and V is a *state valuation*, i.e., a monotonic mapping from the interval $[0, \dots, n-1]$ to finite subsets of $\Pi \setminus \{\bot\}$. Stabilization sequences of length n are also referred to as *n–sequences.* It will be technically convenient to assume that M is *irredundant*, i.e., $V(i-1) \subset V(i)$, for all $0 < i < n$. This assumption, however, is not important for the results in this report.

Definition 3.1 (Sequence Model). *Let $M = (n, V)$ be a stabilization sequence and $C \in PC$. Then, M is said to be a* sequence model *of C, if $M \models C$, where the satisfaction relation $\models$ is defined along the structure of C as follows:*

1. *Always $M \models \mathbf{0}$,*
2. *$M \models C \parallel D$ if $M \models C$ and $M \models D$, and*
3. *$M \models P, \overline{N}/A$ if both $N \cap V(n-1) = \emptyset$ and $P \subseteq V(i)$ imply $A \subseteq V(i)$, for all $i < n$.*

This definition is a shaved version of the standard semantics obtained when reading $C \in \mathsf{PC}$ as an intuitionistic formula [19], i.e., when taking events to be atomic propositions and replacing $\overline{a}$ by negation $\neg a$, concatenation of events and "$\|$" by conjunction "$\wedge$", and the transition slash "/" by implication "$\supset$". An empty trigger, an empty action, and configuration $\mathbf{0}$ are identified with *true*. Then, $M \models C$ if and only if C holds for the intuitionistic Kripke structure M. In the sequel, we abbreviate the set $\{M \mid M \models C\}$ of sequence models of C by $SM(C)$. It will sometimes be useful to consider the sequence models $2SM(C)$ of C of length at most two only, i.e., $2SM(C) =_{\mathrm{df}} \{(n, V) \mid (n, V) \in SM(C) \text{ and } n \leq 2\}$.

In our introductory example, configuration C_{79} is behaviorally equivalent to $C'_{79} =_{\mathrm{df}} \overline{b}/a \parallel b/a$. The latter configuration may be identified with formula $(\neg b \supset a) \wedge (b \supset a)$ which states "*if b is absent throughout a macro step or b is present throughout a macro step, then a is asserted.*" In classic logic, configuration C'_{79} would be deemed equivalent to the single transition $C_{12} = \cdot/a$ corresponding to formula *true* $\supset a$. As mentioned before, this is inadequate as both do not have the same operational behavior, since $C'_{79} \parallel a/b$ fails whereas $C_{12} \parallel a/b$ has step response $\{a, b\}$ in the empty environment. In our intuitionistic semantics, the difference is faithfully witnessed by the 2–sequence $M = (2, V)$, where $V(0) =_{\mathrm{df}} \{a\}$ and $V(1) =_{\mathrm{df}} \{a, b\}$. Here, M is a sequence model of configuration C'_{79} but not of configuration C_{12}.

As another example, consider configurations $\overline{a}/a$ and $\cdot/a$ corresponding to formulas $\neg a \supset a$ and *true* $\supset a$, respectively. In classic logic both are equivalent. Yet, they differ in their operational behavior. The former configuration fails in the empty environment while the latter produces response $\{a\}$. In our intuitionistic semantics, however, both are distinguished: $\neg a \supset a$ specifies "*eventually a must be present*," as $\overline{a}/a$ expects the environment to assert event a in order to avoid failure. This is different from *true* $\supset a$ which specifies "*always a*." Formally, formula $\neg a \supset a$ possesses two sequence models over set $\{a\}$: (i) 2–sequence $(2, V_1)$, where $V_1(0) =_{\text{df}} \emptyset$ and $V_1(1) =_{\text{df}} \{a\}$, and (ii) 1–sequence $(1, V_2)$, where $V_2(0) =_{\text{df}} \{a\}$. However, according to Def. 3.1, $(2, V_1)$ is not a sequence model of formula *true* $\supset a$. Finally, consider formula $(a \supset b) \wedge (b \supset a)$ which corresponds to configuration $a/b \;||\; b/a$. This has also exactly two sequence models over event set $\{a, b\}$: (i) 2–sequence $(2, W_1)$, where $W_1(0) =_{\text{df}} \emptyset$ and $W_1(1) =_{\text{df}} \{a, b\}$, and (ii) 1–sequence $(1, W_2)$ with $W_2(0) =_{\text{df}} \emptyset$. Hence, the environment has to provide at least one event, a or b, in order for response $\{a, b\}$ to occur, i.e., the transitions a/b and b/a cannot mutually trigger each other, in accordance with the principle of causality [20].

Note that the classic semantics is contained in the intuitionistic one by considering 1–sequences only. More precisely, every 1–sequence $M = (1, V)$ may be identified with a Boolean valuation $V' \in \Pi \rightarrow \mathbb{B}$ by taking $V'(a) = tt$ if and only if $a \in V(0)$. Then, $M \models C$ if and only if C classically evaluates to tt under valuation V'. Moreover, it will be convenient to identify a 1–sequence $(1, V)$ with a subset of events, i.e., the set $V(0) \subseteq_{\text{fin}} \Pi \setminus \{\perp\}$. Vice versa, a subset $A \subseteq_{\text{fin}} \Pi \setminus \{\perp\}$ induces the 1–sequence $(1, V)$, where $V(0) =_{\text{df}} A$. Every n–sequence also contains a distinguished classic structure, namely its final state. We refer to the final state of $M = (n, V)$ as M^*, i.e., $M^* = (1, V^*)$ where $a \in V^*(0)$ if and only if $a \in V(n-1)$; sometimes, M^* is simply identified with the final state $V(n-1)$. Finally, we also employ the notation M^i, for $i < n$, to denote the suffix sequence of M that starts at state i, i.e., $M^i =_{\text{df}} (n-i, V^i)$ where $V^i(j) =_{\text{df}} V(i+j)$. It is easy to show that whenever $M \in SM(C)$ then $M^i \in SM(C)$, too.

Proposition 3.2. *Let $C \in PC$ and M be a n–sequence. Then, $M \models C$ implies $M^i \models C$, for all $i < n$.*

As a consequence, one may always construct a model in $2SM(C)$ when given a model in $SM(C)$.

3.2. Characterization of Pnueli and Shalev's Semantics. We now show that the step responses of a parallel configuration C, according to Pnueli and Shalev's semantics, can be characterized as particular sequence models of C, to which we refer as *response models*. The response models of C are those 1–sequence models of C, i.e., subsets $A \subseteq_{\text{fin}} \Pi \setminus \{\perp\}$, that do not occur as the final world of any other sequence model of C except itself. Intuitively, the validity of this characterization is founded in Pnueli and Shalev's closed–world assumption which requires a response to emerge from within the considered configuration and not by interactions with the environment. More precisely, if event set A occurs as the final state of an n–sequence model M, where $n > 1$, then M represents a proper interaction sequence of the considered configuration with its environment that *must* occur in order for C to participate in response A. Hence, if there is no non–trivial n–sequence with $M^* = A$, then C can produce A as an autonomous response.

Definition 3.3 (Response Model). *Let $C \in PC$. Then, $M = (1, V) \in SM(C)$ is a* response model *of C if $K^* = M^*$ implies $K = M$, for all $K \in SM(C)$. The set of response models of C is denoted $RM(C)$.*

Hence, response models of C may be identified with specific classic models of C. Observe, however, that their definition involves essential reference to the intuitionistic semantics of configurations.

Theorem 3.4 (Characterization). *Let $C \in PC$ and $E, A \subseteq_{fin} \Pi$. Then, $C \Downarrow_E A$ if and only if A is a response model of configuration $C \;||\; \cdot/E$.*

Proof. Let us start with a comment concerning our notation for transitions. In this and in the following proofs we will often identify a transition $P, \overline{N}/B$ with the intuitionistic formula $P \wedge \neg N \supset B$. More precisely, formulas P and B stand for the conjunctions of the events in sets P and B, respectively, and formula $\neg N$ abbreviates the conjunction of the negations of all events in set N. This propositional notation reflects precisely our intuitionistic semantics of Def. 3.1. Since $C \Downarrow_E A$ if and only if $(C \,||\, \cdot/E) \Downarrow A$, it suffices to show that $D \Downarrow A$ if and only if A is a response model of D, for all $D \in \mathsf{PC}$ and $A \subseteq_{\mathsf{fin}} \Pi$.

- "$\Longrightarrow$": Let $D \Downarrow A$, and let T be the set of admissible transitions generating response A; in particular, $\bot \notin A$. We show that A is a response model of D. Let us first convince ourselves that A is a model of D, i.e., $A \models D$. Recall that we identify A with the stabilization sequence $(1, V)$, where $V(0) =_{\mathrm{df}} A$. Let $t = P \wedge \neg N \supset B$ be a transition from D. Suppose that $A \models P \wedge \neg N$, i.e., $P \subseteq A$ and $N \cap A = \emptyset$. Since A is the set of events generated from T and since t is enabled by A, we conclude that t must have fired, i.e., $t \in T$. This implies $B \subseteq A$. Thus, $A \models B$, which proves $A \models t$. Since t was arbitrary, A validates all (parallel) transitions of D, whence $A \models D$, as desired.
  Next we show that A is in fact a response model, i.e., there exists no non–classic irredundant extension of A that is a model of D. Suppose $K = (n, V)$ is such an irredundant n–sequence model of D with $K^* = A$ and $K \models D$. If $n = 1$, then $K = A$, and we are done. Otherwise, if $n \geq 2$, the sequence K has at least two states; in particular, we must have $V(n-2) \subset A$. Sequence model K has the following useful properties:

  (1) $\forall b \in \Pi.\ A \models \neg b$ implies $K \models \neg b$, i.e., A and K have the same negated truths.
  (2) $\exists a \in \Pi.\ A \models a$ but $K \not\models a$.

  Prop. (1) implies that K satisfies the negative triggers of all transitions that have fired to produce A, since those are all valid in A and, hence, must be valid in K. Now, we use the fact that if T is the set of transitions — or, more precisely, their corresponding formulas — which have fired to produce A and if $\neg R$ are the cumulated negative triggers, then $T \wedge \neg R \models A$ is a valid consequence in intuitionistic logic. This can be shown without difficulties as an auxiliary lemma, using essentially the deductive nature of the step semantics, e.g., by induction on the number of iterations of the step–construction procedure. Thus, (i) $T \wedge \neg R \models A$, (ii) $K \models T$, since it is a model of D, and (iii) $K \models \neg R$, whence $K \models A$. But this contradicts Prop. (2).
- "$\Longleftarrow$": Suppose M is a response model of D. We must prove $D \Downarrow M$. To this end, consider the set T_M of all (parallel) transitions of D that are enabled in M. We show that

  (1) T_M is an admissible set of transitions in D and
  (2) $\mathsf{act}(T_M) = M$.

  Note that it is clear that $\bot \notin M$, as M is a sequence model. Regarding Prop. (2), take any $t \in T_M$, say $t = P \wedge \neg N \supset B$. Since trigger $P \wedge \neg N$ of t is valid in M and since M is a model of D, we must have $M \models B$, whence $B \subseteq M$. Thus, $\mathsf{act}(T_M) \subseteq M$. For the other inclusion, suppose there exists some $a \in M$ which does not appear as an action of any transition in T_M. We claim, then, that we can extend M to an irredundant 2–sequence model K of D with $K^* = M$. To obtain such a K, take $K =_{\mathrm{df}} (2, V)$, where $V(1) =_{\mathrm{df}} M$ and $V(0) =_{\mathrm{df}} M \setminus \{a\}$. Now, we show that K is a model of D. Take any transition t of D, say $P \wedge \neg N \supset B$. For establishing $K \models t$, we follow the semantic definition of transitions (cf. Def. 3.1). Suppose $i \in \{0, 1\}$, $V(1) \cap N = \emptyset$, and $P \subseteq V(i)$. We have to show that $B \subseteq V(i)$. Since $V(1) = M$ and $M \models t$, this follows immediately in case $i = 1$. So, consider $i = 0$. The assumptions $P \subseteq V(0) \subset V(1)$ and $V(1) \cap N = \emptyset$ mean that t is enabled in $M = V(1)$, whence $t \in T_M$ by construction. But then $a \notin B$, since all events in B are actions of T_M and since a does by assumption not appear as an action in T_M. Now, $a \notin B$ finally means $B = B \setminus \{a\} \subseteq M \setminus \{a\} = V(0)$. Hence, $B \subseteq V(0)$, as desired. This completes the proof that K is a model of t, for arbitrary $t \in \mathsf{trans}(D)$, whence $K \models D$. Consequently, we have extended M to an irredundant sequence model K of D of length 2, which contradicts the assumption that M is a response model. Thus, $M \subseteq \mathsf{act}(T_M)$, and, putting our results together, $M = \mathsf{act}(T_M)$.

Regarding Prop. (1), it is not difficult to prove that $T_M = \mathsf{enabled}(D, \emptyset, T_M)$. Let $t \in T_M$. We claim that t is enabled by the set of actions of T_M. Since, by Prop. (2), M is exactly the set of all actions generated by T_M and since t is enabled in M, transition t must be enabled by T_M. Hence, $T_M \subseteq \mathsf{enabled}(D, \emptyset, T_M)$. Vice versa, let t be a transition of D enabled in T_M, whence enabled in M. Then, $t \in T_M$ by definition. This proves the first part of admissibility. It remains to be shown that there exists some $t \in T_M \setminus T$ such that $t \in \mathsf{enabled}(D, \emptyset, T)$, for any $T \subset T_M$. Let $T \subset T_M$ be a proper subset of T_M. Consider the set $\mathsf{act}(T)$ of actions generated from T, which satisfies $\mathsf{act}(T) \subseteq \mathsf{act}(T_M) = M$ by Prop. (2). We distinguish two cases. First, if $\mathsf{act}(T) = M$, then by definition all transitions in T_M are enabled by $\mathsf{act}(T)$. Thus, since $T_M \setminus T$ is non–empty, there exists at least one transition in T_M outside of T that is enabled by T. Second, assume $\mathsf{act}(T) \subset M$ is a proper subset. We then define the irredundant stabilization sequence $K =_{\mathrm{df}} (2, V)$ as a model extension of M, such that $V(0) =_{\mathrm{df}} \mathsf{act}(T)$ and $V(1) =_{\mathrm{df}} M$. Since $M = K^*$ is a response model by assumption, K cannot be a model of D. Thus, there exists some transition t, say $P \wedge \neg N \supset A$, in D such that $K \not\models t$. By the semantic definition for transitions (cf. Def. 3.1) this means that there exists an $i \in \{0, 1\}$ such that (i) $P \subseteq V(i)$, (ii) $V(1) \cap N = \emptyset$, and (iii) $A \not\subseteq V(i)$. Since $P \subseteq V(i) \subseteq V(1)$ and $V(1) \cap N = \emptyset$, transition t is enabled in $M = V(1)$. Thus, $t \in T_M$. The remaining fact $A \not\subseteq V(i)$ implies $t \notin T$; otherwise, if $t \in T$ then $A \subseteq \mathsf{act}(T) = V(0)$, which contradicts $A \not\subseteq V(i)$, since $V(0) \subseteq V(i)$, for any i. Hence, $t \in T_M \setminus T$ and $t \in \mathsf{enabled}(D, \emptyset, T)$, as desired.

This completes the proof of Thm. 3.4. □

Thm. 3.4 provides a simple model–theoretic characterization of step responses. For example, recall that configuration $\overline{a}/a$ forces Pnueli and Shalev's step construction procedure to fail. As shown before, the only sequence model of $\overline{a}/a$ of length 1 and using only event a is $(1, V_2)$. But $(1, V_2)$ is not a response model since it is the final world of 2–sequence model $(2, V_1)$. Since $\neg a \supset a$ does not have any response model, transition $\overline{a}/a$ can only fail in the empty environment. As another example, re–visit configuration $a/b \parallel b/a$, for which just sequence $(1, W_2)$ is a response model. Thus, $(a/b \parallel b/a) \Downarrow \emptyset$ is the only response in the empty environment.

3.3. Full Abstraction for Parallel Configurations. Sequence models are not only elegant for characterizing Pnueli and Shalev's semantics, but also lead to a fully–abstract semantics for parallel configurations within parallel contexts.

Theorem 3.5 (Full Abstraction). *For all $C, D \in PC$, the following statements are equivalent:*

1. *$SM(C) = SM(D)$.*
2. *$2SM(C) = 2SM(D)$.*
3. *$(C \parallel R) \Downarrow_E A$ if and only if $(D \parallel R) \Downarrow_E A$, for all $R \in PC$ and $E, A \subseteq_{\mathit{fin}} \Pi$.*
4. *$RM(C \parallel R) = RM(D \parallel R)$, for all $R \in PC$.*

This theorem states that we can completely determine the response behavior of a parallel configuration in arbitrary parallel contexts from its sequence models, or indeed its 1– and 2–sequence models. Hence, sequence models contain precisely the information needed to capture all possible interactions of a parallel configuration within all potential environments. To prove Thm. 3.5, we first establish an auxiliary lemma to show that the set of sequence models of at most length two contains the same information as the set of sequence models of arbitrary length.

LEMMA 3.6. *Let $C, D \in PC$, and let K be a stabilization sequence of arbitrary length such that $K \models C$, $K \not\models D$, and $K^* \models D$. Then, there exists a 2–sequence M with $M \models C$, $M \not\models D$, and $M^* = K^*$.*

Proof. Let configurations C and D and n–sequence $K = (n, W)$ be given as stated in the lemma. Clearly, $n \geq 2$, as $K = K^*$ would be inconsistent with the assumptions $K \not\models D$ and $K^* \models D$. Now, let $0 \leq l \leq n-2$ be the largest l such that $K^l \not\models D$ and $K^{l+1} \models D$. Consider the 2–sequence model $M =_{\text{df}} (2, V)$ where $V(0) =_{\text{df}} W(l)$ and $V(1) =_{\text{df}} W(n-1)$, i.e., M consists of the first and the last state of K^l. Obviously, $M^* = K^*$. We will show that $M \models C$ but $M \not\models D$. We first prove that for every transition t, say $P \wedge \neg N \supset A$,

$$K^l \models t \text{ if and only if } M \models t\,. \tag{3.1}$$

From this our claim follows because parallel configurations C and D are conjunctions of transitions and, moreover, $K^l \models C$ and $K^l \not\models D$. Since $K^* = W(n-1) = V(1) = M^*$ we immediately have

$$K^* \models t \;\text{ if and only if }\; M^* \models t \qquad \text{as well as} \qquad M \models \neg N \;\text{ if and only if }\; K^l \models \neg N\,.$$

By construction, $W(l) = V(0)$, whence they force the same events, in particular for P:

$$P \subseteq W(l) \;\text{ if and only if }\; P \subseteq V(0) \qquad \text{as well as} \qquad A \subseteq W(l) \;\text{ if and only if }\; A \subseteq V(0)\,.$$

Taking all this together implies Statement (3.1). □

On this basis, we are now able to establish Thm. 3.5.

Proof. [Theorem. 3.5] We begin with the equivalence of Statements (1) and (2). It is obvious that $SM(C) = SM(D)$ implies $2SM(C) = 2SM(D)$, as 1– and 2–sequence models are just special sequence models. For the other direction, assume w.l.o.g. that $SM(C) \not\subseteq SM(D)$. Hence, there must exist a stabilization sequence K such that $K \models C$ and $K \not\models D$. In the case $K^* \not\models D$, we obtain $2SM(C) \not\subseteq 2SM(D)$ since $K^* \models C$ and since K^* is a classic structure. In the case $K^* \models D$, we apply Lemma 3.6 which yields a 2–sequence model M satisfying $M \models C$ and $M \not\models D$. Thus, $2SM(C) \not\subseteq 2SM(D)$, too. The equivalence of Statements (3) and (4) is an easy consequence of Thm. 3.4. It remains to establish the equivalence of Statements (2) and (4).

- "$\Longrightarrow$": Suppose that $2SM(C) = 2SM(D)$ and that $R \in \mathsf{PC}$. Then, $A \in RM(C \parallel R)$ implies $A \models C$ and $A \models R$. Since A is a classic sequence model of C, it must be a sequence model of D and, hence, of $D \parallel R$. We claim that A actually is a response model of $D \parallel R$. Suppose it was not. Then, there would exist an irredundant sequence model $K = (n, V)$ of $D \parallel R$ satisfying $n \geq 2$ and $K^* = V(n-1) = A$. Since K is irredundant, it contains the 2–sequence $M = (2, W)$, where $W(0) =_{\text{df}} V(n-2)$ and $W(1) =_{\text{df}} V(n-1)$. By the properties of intuitionistic truth (cf. Def. 3.1), $K \models D \parallel R$ implies $M \models D \parallel R$. Hence, there exists a 2–sequence model M with $M^* = A$ and $M \models D \parallel R$. Since $2SM(C) = 2SM(D)$, this implies $M \models C \parallel R$, contradicting the assumption that A is a response model of $C \parallel R$.
- "$\Longleftarrow$": This proof direction needs slightly more work as it involves the construction of a discriminating context. We start off with the assumption $2SM(C) \neq 2SM(D)$. W.l.o.g., let M be a stabilization sequence of length one or two such that $M \models C$ and $M \not\models D$. Moreover, define $A =_{\text{df}} M^*$. We now distinguish two cases.

1. $\underline{A \not\models D.}$ Consider the context

$$R =_{\mathrm{df}} \| \, \{L(0)/A \mid (n, L) \in \mathit{2SM}(C) \text{ and } L^* = A\} \,.$$

   Observe that R is a parallel composition of *finitely* many transitions as A is finite. Moreover, R is non–empty since $M(0)/A$ is a transition in R. It is immediate that A cannot be a response model of $D \,\|\, R$ because, by assumption, it is not even a model of D. We are done if we can show that $A \in \mathit{RM}(C \,\|\, R)$. Since every transition of R is of the form $L(0)/A$, we have $A \models R$. Also, $A \models C$ holds because $M \models C$ and $A = M^*$. Hence, $A \models C \,\|\, R$. Moreover, it is not difficult to show that there cannot exist a 2–sequence K such that $K^* = A$ and $K \models C \,\|\, R$. If such K would exist, it would have to satisfy $K(0) \subset A$ and $K \models C$. Hence, by construction, transition $K(0)/A$ is a parallel component of R. This means $K \not\models R$, since $K \not\models K(0)/A$, which follows from $K(0) \subseteq K(0)$ and $A \not\subseteq K(0)$. But $K \not\models R$ would be a contradiction to $K \models C \,\|\, R$. This shows that there exists no proper weakening K of A that is still a model of $C \,\|\, R$. Thus, A is a response model of $C \,\|\, R$.
2. $\underline{A \models D.}$ Since $M^* = A$ and $M \not\models D$, this assumption implies that M is irredundant, i.e., it is a 2–sequence with $M(0) \subset M(1)$. In this case, we construct a configuration R such that A is a response model of $D \,\|\, R$ but not of $C \,\|\, R$. Consider an arbitrary stabilization sequence K. We define transitions t_K^M as follows; recall that M is a 2–sequence, whence $M(1) = M^* = A$:

$$t_K^M =_{\mathrm{df}} \begin{cases} K(0)/M(0) & \text{if } K(0) \subseteq M(0) \\ K(0)/M(1) & \text{otherwise.} \end{cases}$$

   Again the sets $K(0)$, $M(0)$, and $M(1)$ are finite. These transitions have the property that $M \models t_K^M$, for all K, and $K \not\models t_K^M$, for all K such that $K(0) \neq M(0)$, $K(0) \neq M(1)$, and $K^* = M(1) = A$. The context configuration R is now formed as

$$R =_{\mathrm{df}} \| \, \{t_L^M \mid L \in \mathit{2SM}(D) \text{ and } L^* = A\} \,.$$

   As before, there is only a finite number of L with $L^* = A$, as A is finite. It follows from the above that $M \models R$ and also $A \models R$. Now we compare the response models of $C \,\|\, R$ and $D \,\|\, R$. Obviously, $A \notin \mathit{RM}(C \,\|\, R)$, since M is irredundant with $M^* = A$, and also $M \models C$ and $M \models R$, whence $M \models C \,\|\, R$. We claim that $A \in \mathit{RM}(D \,\|\, R)$. First of all, $A \models D \,\|\, R$. Now suppose there exists an irredundant stabilization sequence K such that $K^* = A$ and $K \models D \,\|\, R$. We may assume that K has length 2 according to Prop. 3.2. By construction, R then contains transition t_K^M, whence $K \models t_K^M$. However, this is impossible unless $K(0) = M(0)$ or $K(0) = M(1)$. If, however, $K(0) = M(0)$, then $K \not\models D$. This follows from $K^* = M(1) = A$ and the assumption $M \not\models D$, as one can show without difficulties. So, we must have $K(0) = M(1)$. Since $K^* = A = M(1)$ and since K is irredundant, we conclude $K = A$. Thus, there cannot exist a non–trivial weakening of A that is a model of $D \,\|\, R$. Hence, $A \in \mathit{RM}(D \,\|\, R)$, as desired.

This completes the proof of Thm. 3.5. □

3.4. Characterization of Sequence Models. Thm. 3.5 does not mean that every set of stabilization sequences can be obtained from a parallel configuration. In fact, from the model theory of intuitionistic logic it is known that in order to specify arbitrary linear sequences, nested implications are needed [19]. Statecharts configurations, however, only use first–order implications and negations. Therefore, we may expect the semantics of configurations to satisfy additional structural properties due to the limited expressiveness of configuration formulas. In fact, it turns out that the sets $SM(C)$ are closed under *sub–sequences*, *refinement*, and *sequential composition.* These notions are defined as follows:

- The m–sequence $M = (m, V)$ is a *sub-sequence* of the n–sequence $N = (n, W)$, written $M \preceq N$, if there exists a mapping $f : [0, \dots, m-1] \rightarrow [0, \dots, n-1]$ such that $V(i) = W(f(i))$, for all $i < m$, and $V(m-1) = W(n-1)$. Note that f must be strictly monotonic since V and W are strictly increasing. In other words, $M \preceq N$ holds if M is obtained from N by dropping some states while preserving the final state.
- The k–sequence $K = (k, U)$ is a *refinement* of the m–sequence $M = (m, V)$ and the n–sequence $N = (n, W)$, written $K \preceq M \sqcap N$, if there exist mappings $f_M : [0, \dots, k-1] \rightarrow [0, \dots, m-1]$ and $f_N : [0, \dots, k-1] \rightarrow [0, \dots, n-1]$ such that $U(k-1) = V(m-1) = W(n-1)$ and $U(i) = V(f_M(i)) \cap W(f_N(i))$, for $i < k$. Intuitively, $K \preceq M \sqcap N$ holds if M, N, and K have the same final state and if every state of K arises from the intersection of a state from M with one from N.
- Finally, the *sequential composition* of $M = (m, V)$ and $N = (n, W)$, such that $V(m-1) \subset W(0)$, is the sequence $M \, ; N = (m+n, U)$ where $U(i) = V(i)$, for $i < m$, and $U(i) = W(i-m)$, otherwise.

One can easily verify that the set $SM(C)$, for every parallel configuration $C \in \mathsf{PC}$, is closed under sub–sequences, refinement, and sequential composition. In the finite case the converse is also valid, i.e., every finite set of stabilization sequences which is closed under sub–sequences, refinement, and sequential composition is the set of sequence models of some parallel configuration, relative to some fixed finite set of events. However, instead of working with sets of sequence models, we will present an equivalent characterization that is much more compact and that employs simple finite lattice structures which we refer to as *behaviors.*

Definition 3.7 (Behavior). *A* behavior $\mathcal{C}$ *is a pair* $\langle F, I \rangle$, *where* $F \subseteq 2^{\Pi \setminus \{\bot\}}$ *and* I *is a function that maps every* $B \in F$ *to a set* $I(B) \subseteq 2^B$ *of subsets of* B, *such that*

1. *I is monotonic, i.e., $B_1 \subseteq B_2$ implies $I(B_1) \subseteq I(B_2)$,*
2. *$I(B)$ is closed under intersection, i.e., $B_1, B_2 \in I(B)$ implies $B_1 \cap B_2 \in I(B)$, and*
3. *$B \in I(B)$.*

*If $F = \{A\}$, for some $A \subseteq_{fin} \Pi$, then $\mathcal{C}$ is called A–*bounded, *or simply* bounded *if A is understood. Moreover, $\mathcal{C}$ is* directed *if $F \neq \emptyset$ and $\forall B_1, B_2 \in F \; \exists B \in F. \; B_1 \subseteq B$ and $B_2 \subseteq B$.*

Intuitively speaking, the first component F of a behavior $\mathcal{C} = \langle F, I \rangle$ captures the possible final responses of $\mathcal{C}$. For every such final response $B \in F$, the event sets $I(B) \subseteq 2^B$ represent the states of all stabilization sequences of C that end in B. Any strictly increasing sequence that moves only through states $I(B)$ and ends in B is considered a stabilization sequence of the behavior. In case $I(B) = \{B\}$ set B is an autonomous response of $\mathcal{C}$. This interpretation is confirmed below in Lemma 3.9 for those behaviors that are obtained from parallel Statecharts configurations.

It is not difficult to show that the pairs of initial and final states occurring together in the sequence models of $C \in \mathsf{PC}$ induce a behavior. More precisely, the *induced* behavior $Beh(C)$ of C is the pair $\langle F(C), I(C) \rangle$ which is defined as follows:

$$F(C) =_{\mathrm{df}} \{E \subseteq \Pi \mid \exists (n, V) \in SM(C). \; V(n-1) = E\} \quad \text{and}$$
$$I(C)(B) =_{\mathrm{df}} \{E \subseteq B \mid \exists (n, V) \in SM(C). \; V(0) = E \text{ and } V(n-1) = B\}.$$

From the property of sub–sequence closure we know that the initial and final states of any sequence model of C form a 2–sequence model of C. Thus, we can also define behavior $\langle F(C), I(C)\rangle$ directly from $2SM(C)$:

$$X \in F(C) \text{ if and only if } X \in 2SM(C)\,, \text{ and}$$
$$X \in I(C)(Y) \text{ if and only if } (X,Y) \in 2SM(C)\,,$$

where we identify a 1–sequence $(1,V)$ with the subset $V(0)$ and a 2–sequence $(2,V)$ with the pair $(V(0), V(1))$. From our construction it is clear that $Beh(C)$ is uniquely determined by $SM(C)$ or, in fact, by $2SM(C)$.

LEMMA 3.8. *For $C \in PC$, $Beh(C)$ is a behavior and, if $\bot$ does not occur in C, then $Beh(C)$ is directed.*

Proof. Observe that, for all stabilization sequences (n,V), we have $\bot \notin V(n-1)$ by definition. Hence, $F(C) \subseteq 2^{\Pi\setminus\{\bot\}}$.

First, we show that $I(C)$ is monotonic. Let $B_1, B_2 \in F(C)$ such that $B_1 \subseteq B_2$, and let $E \in I(C)(B_1)$. If $B_1 = B_2$ nothing needs to be shown, i.e., we have $E \in I(C)(B_2)$ trivially. So, suppose $B_1 \subset B_2$. This means that for some $(n,V) \in SM(C)$, both $V(0) = E$ and $V(n-1) = B_1$ hold. We claim that the stabilization sequence $(n+1, W)$ defined by $W(i) =_{\text{df}} V(i)$, for $0 \leq i < n$, and $W(n) =_{\text{df}} B_2$ is a model of C, which then entails $E \in I(C)(B_2)$. To prove $(n+1,W) \in SM(C)$ we proceed by contradiction. Assume that there exists a transition t, say $P \wedge \neg N \supset D$, of C such that $(n+1,W) \not\models t$. This implies that there must exist some $i \leq n$ such that $P \subseteq W(i)$, $N \cap W(n) = \emptyset$, and $D \not\subseteq W(i)$. Since $B_2 \in F(C)$, set B_2 is the final state of a sequence model of C. Thus, by the properties of intuitionistic truth, the singleton sequence B_2 must be a model of C, too. This means that the final state $W(n) = B_2$ of W must satisfy t, i.e., $D \subseteq W(n)$. Hence, $i < n$ and $W(i) = V(i)$. Now, $N \cap B_2 = N \cap W(n) = \emptyset$ and $B_1 \subseteq B_2$ implies $N \cap V(n-1) = N \cap B_1 = \emptyset$. From this we conclude $(n,V) \not\models t$ which contradicts assumption $(n,V) \in SM(C)$. Hence, we have $(n+1,W) \in SM(C)$ and, as a consequence, $W(0) = V(0) = E$ and $W(n) = B_2$, i.e., $E \in I(C)(B_2)$. This completes the proof that $I(C)$ is monotonic.

Second, we verify that $I(C)(B)$ is intersection closed, for all $B \in F(C)$. Let $E_1, E_2 \in I(C)(B)$ and sequences $(n_1, V_1) \in SM(C)$ and $(n_2, V_2) \in SM(C)$ such that $V_1(n_1 - 1) = V_2(n_2 - 1) = B$, $E_1 = V_1(0)$, and $E_2 = V_2(0)$. Consider the 2–sequence $(2,U)$, where $U(0) =_{\text{df}} E_1 \cap E_2$ and $U(1) =_{\text{df}} B$. We claim that $(2,U) \in SM(C)$. Suppose, by way of contradiction, that t is a transition of C, say $P \wedge \neg N \supset D$, for which $(2,U) \not\models t$. Since $B = U(1)$ and $B \in F(C)$, i.e., B is a singleton model of C, we know that $U(1) \models t$. Hence, any violation of t by $(2,U)$ can only occur if $P \subseteq U(0)$, $N \cap U(1) = N \cap B = \emptyset$, and $D \not\subseteq U(0)$. Since $U(0) = E_1 \cap E_2$ it follows that $P \subseteq E_1$ and $P \subseteq E_2$. Furthermore, $D \not\subseteq U(0)$ implies $D \not\subseteq E_i$, for $i = 1$ or $i = 2$. In either case, the fact that $N \cap B = \emptyset$, as B is the final state of (n_i, V_i) for both $i \in \{1,2\}$, implies $(n_i, V_i) \not\models t$ which contradicts our assumption. Thus, $(2,U) \in SM(C)$, as desired. By construction we have $U(0) = E_1 \cap E_2$ and $U(1) = B$, whence $E_1 \cap E_2 \in I(C)(B)$. This completes the proof that $I(C)(B)$ is intersection closed.

Finally, we show that $Beh(C)$ is directed if failure event $\bot$ does not occur in C. Let $E_1, E_2 \in F(C)$, i.e., $V_1(n_1 - 1) = E_1$ and $V_2(n_2 - 1) = E_2$, for some sequence models $(n_1, V_1), (n_2, V_2) \in SM(C)$. Now, consider the 1–sequence $(1,V)$, where $V(0) =_{\text{df}} E_1 \cup E_2 \cup \mathsf{act}(\mathsf{triggered}(C, E_1 \cup E_2)), i.e., E_1 \subseteq V(0)$ and $E_2 \subseteq V(0)$. Note that $V(0) \subseteq \Pi \setminus \{\bot\}$ and that $\bot$ is by assumption not included in any action. Hence, $(1,V)$ is a stabilization sequence. Moreover, $(1,V)$ is clearly a model of each transition of C and, thus, of C. This implies $(1,V) \in SM(C)$ and, further, $V(0) \in F(C)$. It can also be seen that $\mathsf{act}(\mathsf{trans}(C)) \subseteq \Pi \setminus \{\bot\}$ is a classical model of C, whence $F \neq \emptyset$. Thus, $Beh(C)$ is directed, which finishes the proof. □

The relationship between $\mathit{Beh}(C)$ and $\mathit{SM}(C)$ is further clarified by the following lemma which illustrates how $\mathit{Beh}(C)$ may be uniquely determined from $\mathit{SM}(C)$.

LEMMA 3.9. *Let $C \in \mathit{PC}$ be a parallel configuration.*

1. *For every stabilization sequence (n, V), we have $(n, V) \in \mathit{SM}(C)$ if and only if $V(n-1) \in F(C)$ and $V(i) \in I(C)(V(n-1))$, for all $i < n$.*
2. *$B \in \mathit{RM}(C)$ if and only if $B \in F(C)$ and $I(C)(B) = \{B\}$.*

According to Part (1), a stabilization sequence M is an element of $\mathit{SM}(C)$ if and only if it is a sequence of states from $I(C)(B)$ such that $B \in F(C)$ and B is the final state of M. This implies that not only is $\mathit{Beh}(C)$ uniquely determined by $\mathit{SM}(C)$ but also, vice versa, $\mathit{SM}(C)$ is uniquely determined by $\mathit{Beh}(C)$.

Proof. [Lemma 3.9] Part (2) follows immediately from the definition of $\mathit{Beh}(C)$ and $\mathit{RM}(C)$. In addition, direction "$\Longrightarrow$" of Part (1) is trivial as it follows from the definition of $\mathit{Beh}(C)$. To obtain the reverse direction of Part (1), we assume that $V(n-1) \in F(C)$ and $V(i) \in I(C)(V(n-1))$, for all $i < n$. Now, suppose $(n, V) \notin \mathit{SM}(C)$, i.e., there exists a transition t, say $P \wedge \neg N \supset D$, of C satisfying $(n, V) \not\models t$. Let $i < n$ be some index with $P \subseteq V(i)$, $N \cap V(n-1) = \emptyset$, and $D \not\subseteq V(i)$. Note that such an i must exist since (n, V) refutes t. From the assumption $V(i) \in I(C)(V(n-1))$ we infer the existence of a stabilization sequence $(m, W) \in \mathit{SM}(C)$ with $W(0) = V(i)$ and $W(m-1) = V(n-1)$. But this implies $P \subseteq W(0)$, $N \cap W(m-1) = \emptyset$, and $D \not\subseteq W(0)$, which means $(m, W) \not\models t$ in contradiction to $(m, W) \in \mathit{SM}(C)$. Hence, $(n, V) \in \mathit{SM}(C)$, as desired. □

As a consequence of Lemma 3.9, we obtain that $\mathit{Beh}(C)$ contains the same semantic information as $\mathit{SM}(C)$.

THEOREM 3.10 (Characterization). *$\forall C, D \in \mathit{PC}$. $\mathit{Beh}(C) = \mathit{Beh}(D)$ if and only if $\mathit{SM}(C) = \mathit{SM}(D)$.*

Proof. Direction "$\Longleftarrow$" follows immediately from the fact that the behavior of a configuration is derived from its sequence models. The other direction "$\Longrightarrow$" is an implication of Lemma 3.9(1). □

In conjunction with Thm. 3.5, we conclude that equivalence in arbitrary parallel contexts can equally well be decided by behaviors: $\mathit{Beh}(C) = \mathit{Beh}(D)$ if and only if $(C \parallel R) \Downarrow_E A$ is equivalent to $(D \parallel R) \Downarrow_E A$, for all $R \in \mathsf{PC}$ and $E, A \subseteq_{\mathsf{fin}} \Pi$. The advantage of $\mathit{Beh}(C)$ over $\mathit{SM}(C)$ is that the former provides an *irredundant* representation of parallel configurations. Moreover, every finite behavior can be represented exactly. We call a behavior $\mathcal{C} = \langle F, I \rangle$ *A-finite*, for $A \subseteq_{\mathsf{fin}} \Pi$, if $\mathcal{C}$ is uniquely determined by the subsets of A, i.e., $B \in F$ if and only if $B \cap A \in F$, and $X \in I(B)$ if and only if $X \cap A \in I(B \cap A)$. If $\mathcal{C}$ is A–finite, then the *A-restriction* $\mathcal{C}|_A =_{\mathrm{df}} \langle F|_A, I|_A \rangle$, such that $F|_A =_{\mathrm{df}} F \cap 2^A$ and $I|_A(B) = I(B)$, is finite and contains complete information about $\mathcal{C}$. For representation purposes it is convenient to confuse an A–finite behavior $\mathcal{C}$ with its finite restriction $\mathcal{C}|_A$. In a similar vein, we identify an A–bounded behavior $\mathcal{D} = \langle \{A\}, I \rangle$ with the A–finite behavior generated by it, i.e., the uniquely defined behavior $\mathcal{C}$ such that $\mathcal{C}|_A = \mathcal{D}$. We frequently use these implicit restrictions and extensions in our examples without further mention. The exactness of behaviors as models of configurations is now an implication of the following theorem.

THEOREM 3.11 (Completeness). *$\mathcal{C}$ is an A-finite (directed) behavior if and only if there exists a configuration $C \in \mathit{PC}$ over events A (not using failure event $\perp$) such that $\mathcal{C} = \mathit{Beh}(C)$.*

Proof. Direction "$\Longleftarrow$" of Thm. 3.11 is essentially the statement of Lemma 3.8. A–finiteness is a trivial consequence of the fact that the semantics of a configuration only depends on the events mentioned in it. We may thus focus on direction "$\Longrightarrow$".

Let $\mathcal{C} = \langle F, I \rangle$ be an A–finite behavior. We are going to construct a configuration C over events A such that $Beh(C) = \mathcal{C}$. Since $Beh(C) = \langle F(C), I(C) \rangle$ is also A–finite, we can prove $Beh(C) = \mathcal{C}$ simply by establishing that their A–restrictions are identical. Thus, we only need to consider subsets of A, i.e., prove $F \cap 2^A = F(C) \cap 2^A$ and then $I(Y) = I(C)(Y)$ under the additional assumption that $Y \subseteq A$. Moreover, depending on whether $\mathcal{C}$ is directed or not, we can make further assumptions about A. First, if $\mathcal{C}$ is non–directed, then we assume that $\perp \in A$. This is permitted since A–finiteness is not affected by adding $\perp$ to A. Alternatively, if $\mathcal{C}$ is directed, then we may assume $A \in F$. Since $F \neq \emptyset$, the set $F \cap 2^A$ must be non–empty, too, and by directedness must contain a greatest element $A^* \in F \cap 2^A$. Then, $\mathcal{C}$ is also A^*–finite. Thus, if $A \notin F$, we may use A^* instead of A.

Our construction of configuration C uses the following uniform construction of transitions. We associate with every $E \subseteq B \subseteq A$, such that $B \in F$, an event set $(E, B)^* \subseteq \Pi$ defined by

$$(E, B)^* =_{\mathrm{df}} \bigcap \{E' \in I(B) \mid E \subseteq E' \subseteq B\}\,.$$

Note that this intersection is always non–empty since $E \subseteq B \subseteq B$ and $B \in I(B)$, by Prop. (3) of behaviors (cf. Def. 3.7). Intuitively, $(E, B)^*$ is the "best upper approximation" of stabilization sequence $(2, V)$, where $V(0) =_{\mathrm{df}} E$ and $V(1) =_{\mathrm{df}} B$, in $\mathcal{C}$. By construction and by Prop. (2) of behaviors,

$$E \subseteq (E, B)^* \subseteq B \quad \text{as well as} \quad (E, B)^* \in I(B)\,.$$

The left–hand inclusion $E \subseteq (E, B)^*$ becomes an equality $(E, B)^* = E$ precisely if $E \in I(B)$. The right–hand inclusion $(E, B)^* \subseteq B$ is an equality $(E, B)^* = B$ if and only if $I(B) = \{B\}$. Now, we define a configuration $C \in \mathsf{PC}$ from $\mathcal{C}$ as follows:

$$\begin{aligned} C \quad =_{\mathrm{df}} \quad & \| \{(E \cup \overline{(A \setminus B)})/(E, B)^* \mid E \subseteq B \subseteq A \text{ and } B \in F\} \\ & \| \{(B \cup \overline{(A \setminus B)})/(A \setminus B) \mid B \subseteq A \text{ and } B \notin F\}\,. \end{aligned}$$

This is a finite configuration since all sets involved are finite and subsets of A. Observe that if A does not contain $\perp$, then configuration C does not use $\perp$ either. Hence, if $\mathcal{C}$ is directed, then C is $\perp$–free, since by our assumptions $A \in F$ holds which implies $\perp \notin A$. On the other hand, if $\mathcal{C}$ is non–directed, our assumption $\perp \in A$ has the effect that configuration C actually uses event $\perp$ in its transitions.

The claim now is that $Beh(C) = \mathcal{C}$, i.e., $\langle F(C), I(C) \rangle = \langle F, I \rangle$. As discussed above, by A–finiteness, we can restrict ourselves to subsets of A. Moreover, whenever stabilization sequences occur, it suffices by Lemma 3.6 to consider those of at most length two. For convenience, a sequence $(1, V)$ is identified with the redundant sequence $(2, W)$, where $W(0) =_{\mathrm{df}} W(1) =_{\mathrm{df}} V(0)$. For stabilization sequences $(2, V)$, we also write $(V(0), V(1))$.

- We first show $F(C) \cap 2^A = F \cap 2^A$ and start with $F \cap 2^A \subseteq F(C) \cap 2^A$. Suppose $Y \subseteq A$ is such that $Y \notin F(C)$, i.e., there exists no $X \subseteq Y$ with $(X, Y) \in \mathit{2SM}(C)$. In particular, $(Y, Y) \notin \mathit{2SM}(C)$. Hence, there is a transition t in C which is falsified by (Y, Y). If $t = (B \cup \overline{(A \setminus B)})/(A \setminus B)$, for some $B \subseteq A$ and $B \notin F$, we must have $Y = B$, whence $Y \notin F$. In case $t = (E \cup \overline{(A \setminus B)})/(E, B)^*$, for some $E \subseteq B \subseteq A$ and $B \in F$, we obtain $E \subseteq Y$ and $Y \cap (A \setminus B) = \emptyset$ and $(E, B)^* \not\subseteq Y$. The second property $Y \cap (A \setminus B) = \emptyset$ is equivalent to $Y \subseteq B$. Thus, together with the first property, we have $E \subseteq Y \subseteq B$. Now, suppose $Y \in F$. By Prop. (1) of behaviors (monotonicity), $I(Y) \subseteq I(B)$. This implies $(E, B)^* \subseteq (E, Y)^* \subseteq Y$ which would contradict the third property $(E, B)^* \not\subseteq Y$. Hence, $Y \notin F$, as desired. This proves $F \cap 2^A \subseteq F(C) \cap 2^A$.

For the other inclusion $F(C) \cap 2^A \subseteq F \cap 2^A$, suppose $Y \in F(C)$ and $Y \subseteq A$. Thus, $(Y,Y) \in \mathit{2SM}(C)$. If $Y \notin F$, then C contains transition $t = (Y \cup \overline{(A \setminus Y)})/(A \setminus Y)$. But, as one checks without difficulty, $(Y,Y) \not\models t$ which contradicts $(Y,Y) \in \mathit{2SM}(C)$. Since $Y \notin F$, we either have $Y \subset A$, or $Y = A$ and $\bot \in A$ by our assumption. Hence, $Y \in F$ which establishes $F \cap 2^A \subseteq F(C) \cap 2^A$.

- We show $I(C)(Y) = I(Y)$, for all $Y \in F \cap 2^A = F(C) \cap 2^A$. Fix any $Y \in F \cap 2^A$. We first prove the inclusion $I(Y) \subseteq I(C)(Y)$. To this end, let $X \in I(Y)$ be given. We claim that $(X,Y) \in \mathit{2SM}(C)$ which implies $X \in I(C)(Y)$. In order to show that (X,Y) is a 2–sequence model of C it will be convenient to use indices to refer to the states X and Y of this sequence and to use the notation $(V(0), V(1)) =_{\mathrm{df}} (X,Y)$. Now, consider any of the transitions $t = (E \cup \overline{(A \setminus B)})/(E,B)^*$, where $E \subseteq B \subseteq A$ and $B \in F$. We check that $(V(0), V(1)) \models t$ following the definition of our semantics. If $V(1) \cap (A \setminus B) \neq \emptyset$ or, for no $i \in \{1,2\}$, $E \subseteq V(i)$, then we are done immediately. So assume $V(1) \cap (A \setminus B) = \emptyset$ which is the same as $V(1) \subseteq B$, and choose any $i \in \{0,1\}$ such that $E \subseteq V(i)$. Hence, we have $E \subseteq V(i) \subseteq V(1) \subseteq B$. By Prop. (1) of behaviors, $Y = V(1) \subseteq B$ implies $I(Y) \subseteq I(B)$. Furthermore, we have $V(i) \in I(Y)$. In case $i = 0$, this follows from Prop. (3) of behaviors; in case $i = 1$, this is the assumption $X \in I(Y)$. But $I(Y) \subseteq I(B)$ and $V(i) \in I(Y)$ implies $V(i) \in I(B)$. Hence, $V(i)$ is one of the E' in the intersection $(E,B)^* = \bigcap \{E' \in I(B) \mid E \subseteq E' \subseteq B\}$, from which we conclude $(E,B)^* \subseteq V(i)$. This establishes $(V(0), V(1)) \models t$. Now consider any of the other transitions $t = (B \cup \overline{(A \setminus B)})/(A \setminus B)$, for $B \subseteq A$ with $B \notin F$. To show $(V(0), V(1)) \models t$, again, we just need to consider the case $V(1) \cap (A \setminus B) = \emptyset$ or, equivalently, $V(1) \subseteq B$, and any $i \in \{0,1\}$ such that $B \subseteq V(i)$. Then, we have $B \subseteq V(i) \subseteq V(1) \subseteq B$. This yields $Y = V(1) = B$ which is a contradiction to the assumptions $Y \in F$ and $B \notin F$ by the construction of t. Hence, the proof of $(V(0), V(1)) \models t$ is complete. We are thus finished showing $X \in I(C)(Y)$, whence $I(Y) \subseteq I(C)(Y)$. For the other inclusion, $I(C)(Y) \subseteq I(Y)$, let $X \subseteq A$ be given such that $X \notin I(Y)$. We establish $(X,Y) \not\models (X \cup \overline{(A \setminus Y)})/(X,Y)^*$ which is a transition of C, as $Y \in F$ by assumption. But this follows from the fact that $X \subset (X,Y)^*$, because $X \notin I(Y)$, and $(A \setminus Y) \cap Y = \emptyset$. Thus, $(X,Y) \notin \mathit{2SM}(C)$, whence $X \notin I(C)(Y)$.

This completes the proof of Thm. 3.11. □

Summarizing, behaviors $\mathit{Beh}(C)$, for parallel configurations C, yield a very simple model representation of $\mathit{SM}(C)$. For any given $B \in F(C)$, the set $I(C)(B)$ is a finite $(\cap, \subseteq)$ semi–lattice with maximal element B. For every $B' \supseteq B$, the semi–lattice $I(C)(B)$ is a full sub–lattice of $I(C)(B')$. As a simple example, consider the configuration $C =_{\mathrm{df}} bc/a \parallel ac/b \parallel \overline{a}/a \parallel \overline{b}/b \parallel \overline{c}/c$ over events $A = \{a,b,c\}$. Its behavior $\mathit{Beh}(C)$ is A–finite and, when restricted to the relevant events A, may be depicted as in Fig. 3.1. Since $F(C) = \{A\}$ is a singleton set we only have one $(\cap, \subseteq)$ semi–lattice $I(C)(A)$. Moreover, $\mathit{SM}(C)$ is precisely the set of sequences whose worldwise intersection with A are paths in the diagram ending in top element A.

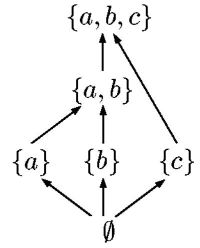

FIG. 3.1. $\{a,b,c\}$–*bounded behavior*

4. Fully–abstract Semantics. We have seen in the previous section that the behavior of a parallel configuration P in all parallel contexts is captured by its set of sequence models $\mathit{SM}(P)$ or, equivalently, its behavior $\mathit{Beh}(P)$. This yields a denotational semantics in which parallel composition is intersection, i.e., $\mathit{SM}(P_1 \parallel P_2) = \mathit{SM}(P_1) \cap \mathit{SM}(P_2)$. Similarly, $\mathit{Beh}(P_1 \parallel P_2) = \mathit{Beh}(P_1) \cap \mathit{Beh}(P_2)$, where the intersection is taken pointwise. The next section shows how this semantics can easily be extended to work with arbitrary contexts, thereby completely characterizing the semantics of PC. However, the question, which still needs to be answered, is how to capture the semantics of the choice operator $+$. In view of the fact that $||$ is logical *conjunction* $\wedge$ in the intuitionistic logic of stabilization sequences, it would be natural to expect that $+$ corresponds to logical *disjunction* $\vee$ over sequence models. Unfortunately, the choice operator $+$ is not a disjunction on sequence models but on behaviors, i.e., on *sets* of sequence models.

As a simple counterexample, for why logical disjunction on sequence models does not suffice, consider transitions a/b and b/a. Moreover, assume that the semantics of $a/b + b/a$ would be completely described by formula $(a \supset b) \vee (b \supset a)$, when interpreted over stabilization sequences, i.e., $\mathit{SM}((a \supset b) \vee (b \supset a)) = \mathit{SM}(a \supset b) \cup \mathit{SM}(b \supset a)$. Now, as one can show, we have $K \models (a \supset b)$ or $K \models (b \supset a)$, for *every* stabilization sequence K. Thus, $\mathit{SM}((a \supset b) \vee (b \supset a))$ contains all stabilization sequences, whence the formula $(a \supset b) \vee (b \supset a)$ is a logical tautology. In terms of sequence models alone, $a/b + b/a$ would be equivalent to the empty configuration $\mathbf{0}$. But obviously both configurations have different response behavior, as, e.g., $(a/b + b/a) \Downarrow_{\{a\}} \{a, b\}$ but only $\mathbf{0} \Downarrow_{\{a\}} \{a\}$. Also, the obvious idea of replacing linear stabilization sequences by arbitrary intuitionistic Kripke models does not work. We will see later that $a/b + b/a$ actually is step congruent to $a/b \;||\; b/a$. Since the formulas $(a \supset b) \vee (b \supset a)$ and $(a \supset b) \wedge (b \supset a)$ are not intuitionistically equivalent, we cannot read $+$ as disjunction on arbitrary Kripke models. It does not appear sensible to try and find a an intermediate class of intuitionistic Kripke models such that the behavior of sum configurations $P_1 + P_2$ can be characterized by the disjunctive formula $P_1 \vee P_2$. Such a semantics would have to use a modified interpretation of transition implication to account for different enabling properties. The next section shows that we need to distinguish transition a/a, which is triggered by a, from transition b/b, which is triggered by b. The naive logical interpretation would identify both transitions with *true*.

Instead of trying to read operator $+$ as logical disjunction, we will use semantic–preserving transformations to eliminate $+$ in favor of parallel composition, whose semantics we already know. There are two methods for achieving this. The naive method is to encode $+$ in terms of $||$ using additional distinguished events to achieve mutual exclusion between the transitions on different sides of the choice operator. This will be discussed in App. C. The other method is to use an expansion law to distribute operator $+$ over operator $||$ and to transform a configuration $C \in \mathsf{C}$ into a standard form $\sum_i C_i$, where all $C_i \in \mathsf{PC}$ are parallel configurations. The semantics of C is then uniquely determined from the semantics of all C_i. The second method will be our main focus in this report since it is more algebraic than the first one and also does not depend on the use of distinguished events.

4.1. Reduction to Parallel Contexts. For extending the full–abstraction result to arbitrary contexts, one must address the following compositionality problem for $+$ which already manifests itself in Pnueli and Shalev's semantics. Consider configurations $C =_{\mathrm{df}} \overline{a}/b$ and $D =_{\mathrm{df}} \overline{a}/b \;||\; a/a$ which have the same responses in all parallel contexts, i.e., $\mathit{Beh}(C) = \mathit{Beh}(D)$. However, in the choice context $\Phi[x] = (\cdot/e + x) \;||\; \cdot/a$, we obtain $\Phi[D] \Downarrow \{a\}$ but $\Phi[C] \not\Downarrow \{a\}$. This context is able to detect that D is enabled by environment $\cdot/a$ while C is not. Hence, to be fully compositional one has to take into account whether there exists a transition in C that is triggered for a set A of events. To store the desired information, we use the *triggering indicator* $\rho(C, A) \in \mathbb{B} =_{\mathrm{df}} \{\mathit{ff}, \mathit{tt}\}$ defined by $\rho(C, A) =_{\mathrm{df}} \mathit{tt}$, if $\mathsf{triggered}(C, A) \neq \emptyset$, and $\rho(C, A) =_{\mathrm{df}} \mathit{ff}$, otherwise. When $C \Downarrow A$, let us call response A *active*, if $\rho(C, A) = \mathit{tt}$, and *passive*, otherwise. This distinction is all we need to reduce step congruence to parallel contexts. Indeed, two configurations are step–congruent if and only if they have the same active and passive step responses in all *parallel* contexts.

Proposition 4.1. *Let $C, D \in \mathsf{C}$. Then, $C \simeq D$ if and only if $\forall P \in \mathsf{PC}, E, A \subseteq_{\mathit{fin}} \Pi, b \in \mathbb{B}$. $(C \;||\; P \Downarrow_E A$ and $\rho(C, A) = b)$ if and only if $(D \;||\; P \Downarrow_E A$ and $\rho(D, A) = b)$.*

This proposition is a corollary to the more general Thm. 4.13 presented in Sec. 4.4. Prop. 4.1 now suggests the following refinement of the naive fully–abstract semantics $[\![\cdot]\!]_0$. For every $C \in C$, we define

$$[\![C]\!]_1^b =_{\mathrm{df}} \{\langle A, P\rangle \mid (C \parallel P) \Downarrow A,\ \rho(C, A) = b,\ P \in \mathsf{PC}\}\,,$$

where $b \in \mathbb{B}$. We may view $[\![C]\!]_1^{tt}$ as the collection of *active* and $[\![C]\!]_1^{f\!f}$ as the collection of *passive* responses for C in *parallel contexts*. From Prop. 4.1, then, we obtain the following result.

PROPOSITION 4.2. *Let $C, D \in C$. Then, $C \simeq D$ if and only if $[\![C]\!]_1^{tt} = [\![D]\!]_1^{tt}$ and $[\![C]\!]_1^{f\!f} = [\![D]\!]_1^{f\!f}$.*

4.2. Reduction to Parallel Configurations. The next step is to eliminate the choice operator from the configurations themselves and to show that the response behavior of every configuration can be determined from that of its parallel components. As mentioned earlier, this will be achieved by transforming configurations into a standard form in which the choice operator is the outermost operator.

To begin with the development of a standard form, please observe that the naive distributivity law $(t_1 + t_2) \parallel t_3 \simeq (t_1 \parallel t_3) + (t_2 \parallel t_3)$, with the two occurrences of t_3 on the right–hand side suitably renamed, does in general not hold. As a counterexample, consider transitions $t_i =_{\mathrm{df}} a_i\overline{b_i}/c_i$, for $1 \leq i \leq 3$, and assume that all events are mutually distinct. Then, in a context in which transition t_2 is enabled but not transition t_1, transition t_3 in $C =_{\mathrm{df}} (t_1 + t_2) \parallel t_3$ is forced to interact with t_2, while in $D =_{\mathrm{df}} (t_1 \parallel t_3) + (t_2 \parallel t_3)$ it may run by itself in the summand $t_1 \parallel t_3$. For example, if $E = \{a_2, a_3\}$ then $D \Downarrow_E \{a_2, a_3, c_3\}$, but the only A with $c_3 \in A$ and $C \Downarrow_E A$ is $A = \{a_2, a_3, c_2, c_3\}$. The same applies if the context enables t_1 but not t_2. The naive distributivity law, however, can be patched as

$$(t_1 + t_2) \parallel t_3 \ \simeq\ t_1 \parallel D_1(t_3) + t_2 \parallel D_2(t_3)\,,$$

where configurations $D_i(t_3)$, for $i \in \{1, 2\}$, are suitable weakenings of t_3 that disable transition t_3, whenever t_i is disabled but t_{3-i} is enabled. There are two ways for defining such configurations.

The most elegant solution is to exploit the failure event $\perp$. In the example, we could define $D_i(t_3) =_{\mathrm{df}} D_i \parallel t_3$, for $i \in \{1, 2\}$, where

$$D_i =_{\mathrm{df}} \overline{a}_i a_{3-i}\overline{b}_{3-i}/\perp \ \parallel\ b_i a_{3-i}\overline{b}_{3-i}/\perp\,.$$

The "watchdog" configuration D_i is enabled exactly if t_i is not enabled and t_{3-i} is, in which case it produces a failure. Formally, for all parallel contexts P, configuration D_i has the property $(D_i \parallel P) \Downarrow A$ if and only if (i) $P \Downarrow A$ and (ii) A triggers t_i or does not trigger t_{3-i}. Thus, D_i does not change any of the responses of P, it only prohibits some of them. We will see below how this can be generalized, namely how one may construct, for any given configurations C_1 and C_2, a watchdog configuration $\mathsf{watch}(C_1, C_2)$ such that $(D \parallel \mathsf{watch}(C_1, C_2)) \Downarrow A$ if and only if $D \Downarrow A$ and $\mathsf{triggered}(C_1, A) \neq \emptyset$ or $\mathsf{triggered}(C_2, A) = \emptyset$.

The second method of patching the naive distributivity law is to modify the parallel context itself and to strengthen the triggers of all its transitions. In our example, $D_i(t_3)$ would modify transition t_3 rather than composing a watchdog parallel to it. Appropriate weakenings $D_i(t_3)$, for $i \in \{1, 2\}$, satisfying $C \simeq t_1 \parallel D_1(t_3) + t_2 \parallel D_2(t_3)$ are

$$D_i(t_3) =_{\mathrm{df}} a_i\overline{b}_i a_3\overline{b}_3/c_3 \parallel \overline{a}_{3-i}a_3\overline{b}_3/c_3 \parallel b_{3-i}a_3\overline{b}_3/c_3\,.$$

Now, configuration $D_i(t_3)$ has the same action as t_3, but is only enabled when t_3 is *and* when t_i is enabled or t_{3-i} is disabled. As intuitionistic formula, $D_i(t_3)$ is equivalent to $((a_i \wedge \neg b_i) \vee \neg a_{3-i} \vee b_{3-i}) \supset t_3$. This is the formal weakening of t_3 by the extra precondition $(a_i \wedge \neg b_i) \vee \neg a_{3-i} \vee b_{3-i}$ which captures exactly the situations in which t_i is enabled or t_{3-i} is disabled. This second approach, which does not depend of the use of explicit $\perp$ events in the actions of transitions, can be generalized to arbitrary configurations. Since this method is less local and more tedious, we do not consider it further. Both methods are essentially equivalent in the sense that the first version of $D_i(t_3)$, i.e., $\mathsf{watch}(t_i, t_{3-i}) \parallel t_3 = \overline{a}_i a_{3-i} \overline{b}_{3-i}/\perp \parallel b_i a_{3-i} \overline{b}_{3-i}/\perp \parallel t_3$, is step congruent to the second version, i.e., $((a_i \wedge \neg b_i) \vee \neg a_{3-i} \vee b_{3-i}) \supset t_3 = a_i \overline{b}_i a_3 \overline{b}_3/c_3 \parallel \overline{a}_{3-i} a_3 \overline{b}_3/c_3 \parallel b_{3-i} a_3 \overline{b}_3/c_3$, as can be derived from our intuitionistic semantics. Hence, the use of the failure event $\perp$ in the watchdog configurations is inessential.

To formally construct watchdogs in a finitary fashion, we need to refer to the events that occur in a configuration. For every configuration C, let $\Pi(C)$ denote the set of all events that syntactically occur in C. Then, we define $\mathsf{watch}(C_1, C_2) \in \mathsf{PC}$ to be the parallel configuration

$$\parallel \{A, \overline{E \setminus A}/\perp \mid A \subseteq E = \Pi(C_1) \cup \Pi(C_2),\ \rho(C_1, A) = \mathit{ff},\ \text{and}\ \rho(C_2, A) = \mathit{tt}\}\,.$$

The crucial semantic property of watchdogs is now stated in the following proposition.

Proposition 4.3. *Let $C_1, C_2, D \in \mathsf{C}$. Then, $(D \parallel \mathsf{watch}(C_1, C_2)) \Downarrow A$ if and only if $D \Downarrow A$ and $\mathsf{triggered}(C_1, A) \neq \emptyset$ or $\mathsf{triggered}(C_2, A) = \emptyset$.*

Proof. In the following, let $E =_{\mathrm{df}} \Pi(C) \cup \Pi(D)$. We begin with direction "$\Longrightarrow$". Since all transitions of $\mathsf{watch}(C_1, C_2)$ have event $\perp$ as their only action event, it follows from $(D \parallel \mathsf{watch}(C_1, C_2)) \Downarrow A$ that none of the watchdog transitions can be enabled. This implies that response A must come from configuration D alone, i.e., $D \Downarrow A$. In particular, transition $A, \overline{E \setminus A}/\perp$ cannot be included in $\mathsf{watch}(C_1, C_2)$; otherwise, it would be enabled by response $A \subseteq E$. But this implies $\rho(C_1, A) \neq \mathit{ff}$ or $\rho(C_2, A) \neq \mathit{tt}$, or, equivalently, $\mathsf{triggered}(C_1, A) \neq \emptyset$ or $\mathsf{triggered}(C_2, A) = \emptyset$.

For proving direction "$\Longleftarrow$", let us assume (1) $D \Downarrow A$ and (2) $\mathsf{triggered}(C_1, A) \neq \emptyset$ or $\mathsf{triggered}(C_2, A) = \emptyset$. Now, given any event set $A' \subseteq E$ satisfying $\rho(C_1, A') = \mathit{ff}$ and $\rho(C_2, A') = \mathit{tt}$, Assumption (2) implies $A \neq A'$. This means that none of the transitions $A', \overline{E \setminus A'}/\perp$ of $\mathsf{watch}(C_1, C_2)$ is enabled. Therefore, Assumption (1) implies $(D \parallel \mathsf{watch}(C_1, C_2)) \Downarrow A$ by the definition of step responses. □

The watchdogs admit the following simple expansion law whose proof, which can be found in App. B, is a direct application of Prop. 4.1.

Lemma 4.4 (Expansion). *Let $P, Q, R \in \mathsf{C}$. Then, $(P + Q) \parallel R \simeq (\mathsf{watch}(P, Q) \parallel P \parallel R) + (\mathsf{watch}(Q, P) \parallel Q \parallel R)$.*

Repeated application of Lemma 4.4 (expansion) can be used to systematically push all occurrences of choice operator $+$ to the outside of the configuration C under consideration, until $+$ becomes the outermost operator. We can think of this transformation of C as a static analysis which reveals the top–level choice structure of C. The general expansion algorithm, which is omitted here, associates with every $C \in \mathsf{C}$ a set $\mathsf{ind}(C)$ of indices and, for every $i \in \mathsf{ind}(C)$, a parallel configuration $C_i \in \mathsf{PC}$. The configurations C_i essentially correspond to the maximal consistent subsets of $\mathsf{trans}(C)$, patched up with appropriate watchdog configurations.

Lemma 4.5 (Standard Form). *Let $C \in \mathsf{C}$. Then, there exists a finite index set $\mathsf{ind}(C)$ and parallel configurations $C_i \in \mathsf{PC}$, for $i \in \mathsf{ind}(C)$, such that $C \simeq \sum_{i \in \mathsf{ind}(C)} C_i$.*

Hence, $[\![C]\!]_1^b = [\![\sum_{i \in \mathsf{ind}(C)} C_i]\!]_1^b$ by Prop. 4.2, for $b \in \mathbb{B}$. Moreover, since an active response of a sum must be an active response of *one* of its summands and since a passive response of a sum always is a passive response of *all* of its summands, we have

$$\llbracket \sum_{i \in \mathsf{ind}(C)} C_i \rrbracket_1^{tt} = \bigcup_{i \in \mathsf{ind}(C)} \llbracket C_i \rrbracket_1^{tt} \quad \text{and} \quad \llbracket \sum_{i \in \mathsf{ind}(C)} C_i \rrbracket_1^{\mathit{ff}} = \bigcap_{i \in \mathsf{ind}(C)} \llbracket C_i \rrbracket_1^{\mathit{ff}}.$$

Thus, we obtain the following proposition which states the desired reduction of the full–abstraction problem to parallel configurations within parallel contexts.

PROPOSITION 4.6. *Let $C, D \in \mathsf{C}$. Then, $C \simeq D$ if and only if*

$$\bigcup_{i \in \mathit{ind}(C)} \llbracket C_i \rrbracket_1^{tt} = \bigcup_{j \in \mathit{ind}(D)} \llbracket D_j \rrbracket_1^{tt} \quad \textit{and} \quad \bigcap_{i \in \mathit{ind}(C)} \llbracket C_i \rrbracket_1^{\mathit{ff}} = \bigcap_{j \in \mathit{ind}(D)} \llbracket D_j \rrbracket_1^{\mathit{ff}}.$$

Proof. We present the proof for two indices only, i.e., we assume $C \simeq C_1 + C_2$ and $D \simeq D_1 + D_2$, where $C_i, D_j \in \mathsf{PC}$ are parallel configurations. The general case is handled in the same way, noting that $+$ is associative. Observe that the statement of Prop. 4.6 reduces to the congruence condition $\llbracket C \rrbracket_1^b = \llbracket D \rrbracket_1^b$ expressed in Prop. 4.2, in case both configurations have only one index. In what follows, $\rho(C, A) \in \mathbb{B}$ denotes again the enabling indicator, so that $\rho(C, A) = \mathit{ff}$, if $\mathsf{triggered}(C, A) = \emptyset$, and $\rho(C, A) = \mathit{tt}$, otherwise.

- "$\Longleftarrow$": We assume

  $$\begin{aligned} \llbracket C_1 \rrbracket_1^{tt} \cup \llbracket C_2 \rrbracket_1^{tt} &= \llbracket D_1 \rrbracket_1^{tt} \cup \llbracket D_2 \rrbracket_1^{tt} \quad \text{and} && (4.1) \\ \llbracket C_1 \rrbracket_1^{\mathit{ff}} \cap \llbracket C_2 \rrbracket_1^{\mathit{ff}} &= \llbracket D_1 \rrbracket_1^{\mathit{ff}} \cap \llbracket D_2 \rrbracket_1^{\mathit{ff}}. && (4.2) \end{aligned}$$

  We must show, by Prop. 4.1, that for every parallel configuration $P \in \mathsf{PC}$ and every $E, A \subseteq_{\mathsf{fin}} \Pi$,

  $$((C_1 + C_2) \parallel P) \Downarrow_E A \ \text{ implies } \ ((D_1 + D_2) \parallel P) \Downarrow_E A \text{ and } \rho(D_1 + D_2, A) = \rho(C_1 + C_2, A) \quad (4.3)$$

  and vice versa, with the roles of C_i and D_i interchanged. We may assume that $E = \emptyset$ since any E is already quantified implicitly by P. Moreover, it suffices to prove the implication in Statement (4.3) because of symmetry. Suppose that $((C_1 + C_2) \parallel P) \Downarrow A$. By Lemma 4.9 we have to consider the following two cases:

  1. There exists some index $i \in \{1, 2\}$ such that $(C_i \parallel P) \Downarrow A$ and $\rho(C_i, A) = \mathit{tt}$.
  2. For both indices $i \in \{1, 2\}$, it is true that $\rho(C_i, A) = \mathit{ff}$ and $(C_i \parallel P) \Downarrow A$.

  In Case (1), by definition, $\langle A, P \rangle \in \llbracket C_i \rrbracket_1^{tt}$. From Equation (4.1) it follows that there exists some $j \in \{1, 2\}$ such that $\langle A, P \rangle \in \llbracket D_j \rrbracket_1^{tt}$. But this yields $((D_1 + D_2) \parallel P) \Downarrow A$ and $\rho(D_j, A) = \mathit{tt}$ when reading Lemma 4.9 backwards. Hence, $\rho(D_1 + D_2, A) = \mathit{tt} = \rho(C_1 + C_2, A)$ which proves Statement (4.3) in Case (1). Regarding Case (2), $\langle A, P \rangle \in \llbracket C_i \rrbracket_1^{\mathit{ff}}$ holds, whence by Equation (4.2), $\langle A, P \rangle \in \llbracket D_j \rrbracket_1^{\mathit{ff}}$, for both $j \in \{1, 2\}$. This means that $(D_1 \parallel P) \Downarrow A$ and $(D_2 \parallel P) \Downarrow A$, as well as $\rho(D_1, A) = \mathit{ff} = \rho(D_2, A)$. So, by employing Lemma 4.9 backwards, we obtain $((D_1 + D_2) \parallel P) \Downarrow A$ and $\rho(D_1 + D_2, A) = \mathit{ff} = \rho(C_1 + C_2, A)$, as desired.
- "$\Longrightarrow$": For this direction, let us assume $C \simeq D$, i.e., $C_1 + C_2 \simeq D_1 + D_2$. Let $\langle A, P \rangle \in \llbracket C_1 \rrbracket_1^{tt}$, i.e., $(C_1 \parallel P) \Downarrow A$ and $\rho(C_1, A) = \mathit{tt}$. By Lemma 4.9, then, $((C_1 + C_2) \parallel P) \Downarrow A$, from which we may infer $((D_1 + D_2) \parallel P) \Downarrow A$ and $\rho(D_1 + D_2, A) = \rho(C_1 + C_2, A) = \mathit{tt}$ by Prop. 4.1. This means, by Lemma 4.9, that $\rho(D_i, A) = \mathit{tt}$ and $(D_i \parallel P) \Downarrow A$, for some $i \in \{1, 2\}$. Thus, $\langle A, P \rangle \in \llbracket D_1 \rrbracket_1^{tt} \cup \llbracket D_2 \rrbracket_1^{tt}$ which implies $\llbracket C_1 \rrbracket_1^{tt} \subseteq \llbracket D_1 \rrbracket_1^{tt} \cup \llbracket D_2 \rrbracket_1^{tt}$. A similar argument shows that $\llbracket C_2 \rrbracket_1^{tt} \subseteq \llbracket D_1 \rrbracket_1^{tt} \cup \llbracket D_2 \rrbracket_1^{tt}$ which yields $\llbracket C_1 \rrbracket_1^{tt} \cup \llbracket C_2 \rrbracket_1^{tt} \subseteq \llbracket D_1 \rrbracket_1^{tt} \cup \llbracket D_2 \rrbracket_1^{tt}$. The other direction follows by symmetry.

Finally, assume $\langle A, P\rangle \in [\![C_1]\!]_1^{\mathit{ff}} \cap [\![C_2]\!]_1^{\mathit{ff}}$, i.e., $\rho(C_i, A) = \mathit{ff}$ and $(C_i \parallel P) \Downarrow A$. Lemma 4.9 implies $((C_1 + C_2) \parallel P) \Downarrow A$. Now we apply Prop. 4.1 again, which establishes $((D_1 + D_2) \parallel P) \Downarrow A$. Moreover, $\rho(D_1 + D_2, A) = \rho(C_1 + C_2, A) = \mathit{ff}$, whence both $\rho(D_1, A) = \mathit{ff} = \rho(D_2, A)$. A final reference to Lemma 4.9 implies $(D_1 \parallel P) \Downarrow A$ and $(D_2 \parallel P) \Downarrow A$. This verifies the inclusion $[\![C_1]\!]_1^{\mathit{ff}} \cap [\![C_2]\!]_1^{\mathit{ff}} \subseteq [\![D_1]\!]_1^{\mathit{ff}} \cap [\![D_2]\!]_1^{\mathit{ff}}$. The other direction, again, is by symmetry.

This finishes the proof. □

Prop. 4.6 yields a second refinement of our fully–abstract semantics that now only depends on the response behavior of parallel configurations in parallel contexts. However, it still refers to the syntax. In the next section the main work will be done, presenting a semantic analysis of the dynamic interaction between parallel configurations.

We finally want to remark that our definition of watchdogs is not the most efficient one possible. For instance, consider configurations $C_1 =_{\mathrm{df}} a/bc$ and $C_2 =_{\mathrm{df}} \overline{b}/cd$, for which $E =_{\mathrm{df}} \Pi(C_1) \cup \Pi(C_2) = \{a, b, c, d\}$. Then, the sets $A \subseteq E$, for which both $\rho(C_1, A) = \mathit{ff}$ and $\rho(C_2, A) = \mathit{tt}$, are $A_1 =_{\mathrm{df}} \emptyset$, $A_2 =_{\mathrm{df}} \{c\}$, $A_3 =_{\mathrm{df}} \{d\}$, and $A_4 =_{\mathrm{df}} \{c, d\}$. Thus, we get by our definition of watchdogs:

$$\begin{aligned}\mathsf{watch}(C_1, C_2) &= A_1, \overline{E \setminus A_1}/\bot \parallel A_2, \overline{E \setminus A_2}/\bot \parallel A_3, \overline{E \setminus A_3}/\bot \parallel A_4, \overline{E \setminus A_4}/\bot \\ &= \overline{a}\,\overline{b}\,\overline{c}\,\overline{d}/\bot \parallel c\,\overline{a}\,\overline{b}\,\overline{d}/\bot \parallel d\,\overline{a}\,\overline{b}\,\overline{c}/\bot \parallel c\,d\,\overline{a}\,\overline{b}/\bot\,.\end{aligned}$$

This parallel configuration, read as logic formula, corresponds to the conjunction

$$\neg(\neg a \wedge \neg b \wedge \neg c \wedge \neg d) \;\wedge\; \neg(c \wedge \neg a \wedge \neg b \wedge \neg d) \;\wedge\; \neg(d \wedge \neg a \wedge \neg b \wedge \neg c) \;\wedge\; \neg(c \wedge d \wedge \neg a \wedge \neg b)$$

which is classically, as well as intuitionistically, equivalent to $\neg a \wedge \neg b$. Hence, the four sets A_1–A_4 could as well be described by the conjunction $\neg a \wedge \neg b$. Indeed, one can show that $\mathsf{watch}(C_1, C_2) \simeq \overline{a}\,\overline{b}/\bot$ which is obviously a more compact formulation. In general, as suggested above, we may invoke classic Boolean logic to simplify watchdogs, as watchdogs are essentially negated formulas which behave classically.

4.3. Full–abstraction Theorem. Once a configuration $C \in \mathsf{C}$ is transformed into a sum $\sum_{i \in \mathsf{ind}(C)} C_i$ of parallel configurations, its semantics may be uniquely determined by the behavior of the $C_i \in \mathsf{PC}$. By Prop. 4.1 it is enough to know the responses of each C_i in all parallel contexts, together with the information of whether these responses are active or passive. Moreover, by Thms. 3.5 and 3.10, the responses of C_i in all parallel contexts are characterized by their behaviors $\mathit{Beh}(C_i)$. Therefore, the responses of C in all contexts must be determined by the sets $\mathit{Beh}(C_i)$ and $\rho(C_i, A)$, for all $i \in \mathsf{ind}(C)$ and $A \subseteq_{\mathsf{fin}} \Pi \setminus \{\bot\}$. Unfortunately, the obvious but somewhat naive idea of simply collecting all the sets $\mathit{Beh}(C_i)$, together with their triggering behavior $\lambda A.\, \rho(C_i, A)$, and then considering the identity of sets as equivalence does not work. The semantics defined in this direct way, namely $[\![C]\!] =_{\mathrm{df}} \{\langle\, \mathit{Beh}(C_i)\,, \lambda A.\, \rho(C_i, A)\,\rangle \mid i \in \mathsf{ind}(C)\}$, would not allow us to derive, e.g., the congruence $a/b + b/a \simeq a/b \parallel b/a$. Indeed, it is not the case that $[\![a/b + b/a]\!] = \{\langle \mathit{Beh}(a/b), \lambda A.\, \rho(a/b, A)\rangle\,, \langle \mathit{Beh}(b/a), \lambda A.\, \rho(b/a, A)\rangle\}$ is the same set as $[\![a/b \parallel b/a]\!] = \{\langle \mathit{Beh}(a/b \parallel b/a), \lambda A.\, \rho(a/b \parallel b/a, A)\rangle\}$ since, e.g., $\mathit{Beh}(a/b)$ is different from $\mathit{Beh}(a/b \parallel b/a)$. However, it is true that $\mathit{Beh}(a/b)$ and $\mathit{Beh}(b/a)$ together cover the same behavior as $\mathit{Beh}(a/b \parallel b/a)$. To achieve a simple formalization of this covering property, it is useful to consider the "complements" of $\mathit{Beh}(a/b)$, $\mathit{Beh}(b/a)$, and $\mathit{Beh}(a/b \parallel b/a)$, to which we refer as (semantic) *contexts.*

Definition 4.7 (Context). *Let $A \subseteq_{\mathit{fin}} \Pi$. An A-bounded behavior $\mathcal{P} = \langle F, I\rangle$ is called an A-*context *for $C \in \mathit{PC}$ if (i) $A \in F(C)$, and (ii) $I(A) \cap I(C)(A) = \{A\}$ holds, where $\langle F(C), I(C)\rangle = \mathit{Beh}(C)$.*

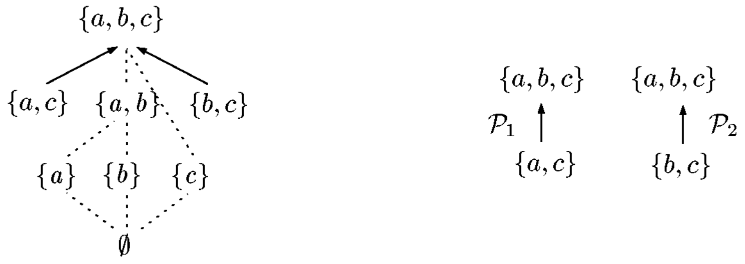

FIG. 4.1. *Complement behavior for Fig. 3.1 (left) and its covering $\{a,b,c\}$–contexts (right).*

An A–context $\mathcal{P}$ of C represents a set of sequences that all end in the final world A, in which also some sequence model of C must end (cf. Prop. (i)), but which only have the final world A in common with the sequence models of C (cf. Prop. (ii)). These properties imply that, for every configuration P with $\mathit{Beh}(P) = \mathcal{P}$, we have $(C \parallel P) \Downarrow A$. Note that, since every A–context $\langle F, I\rangle$ is A–bounded, I is essentially just a $\cap$–closed subset of 2^A with top element A. In other words, an A–context $\langle F, I\rangle$ may be identified with the complete $(\cap, \subseteq)$ sub–semi–lattice $I(A)$ of 2^A. We will henceforth use the simpler presentation $\langle A, I(A)\rangle$ rather than $\langle\{A\}, I\rangle$. In fact, we might even write $I(A)$ since the top element is uniquely determined, but it is often useful to indicate the top element explicitly.

In the following, we will only be interested in the maximal A–contexts of a configuration $C \in \mathsf{PC}$, where maximality is with respect to the natural component–wise subset–ordering on A–bounded behaviors. More precisely, given two A–bounded behaviors $\mathcal{P} = \langle A, I(A)\rangle$ and $\mathcal{P}' = \langle A, I'(A)\rangle$, we say that $\mathcal{P}$ is a *sub–behavior* of $\mathcal{P}'$, written $\mathcal{P} \subseteq \mathcal{P}'$, if $I(A) \subseteq I'(A)$. Then, an A–context of C is called *maximal* if $\mathcal{P} \subseteq \mathcal{P}'$ implies $\mathcal{P} = \mathcal{P}'$, for all A–contexts $\mathcal{P}'$ of C. Because of the finiteness of A–bounded behaviors, every A–context of C must be contained in a maximal one.

Consider again the example $C =_{\mathrm{df}} bc/a \parallel ac/b \parallel \overline{a}/a \parallel \overline{b}/b \parallel \overline{c}/c$ from above, whose A–bounded behavior $\mathit{Beh}(C) = \langle\{A\}, I(C)\rangle$, where $A = \{a,b,c\}$, is described by the diagram of Fig. 3.1. To get the A–contexts of C, we must consider the "holes" in $I(C)(A)$, i.e., all $B \subset A$ that are missing in the lattice of Fig. 3.1. This is illustrated by the left diagram in Fig. 4.1, where lattice $\mathit{Beh}(C)$ is indicated by dashed lines and the holes by solid arrows. As one can see, this "complement" is not itself a behavior, e.g., it is not $\cap$–closed, but it can be covered by the two A–contexts

$$\mathcal{P}_1 =_{\mathrm{df}} \langle\{a,b,c\}, \{\{a,c\}\,,\,\{a,b,c\}\}\rangle \quad \text{and} \quad \mathcal{P}_2 =_{\mathrm{df}} \langle\{a,b,c\}, \{\{b,c\}\,,\,\{a,b,c\}\}\rangle\,, \tag{4.4}$$

which are drawn separately in Fig. 4.1 on the right. In fact, $\mathcal{P}_1$ and $\mathcal{P}_2$ are the two maximal A–contexts of $\mathit{Beh}(C)$. Since they are behaviors, the A–contexts can be represented by parallel configurations, such as $P_1 =_{\mathrm{df}} \cdot/ac \parallel \overline{b}/b$ and $P_2 =_{\mathrm{df}} \cdot/bc \parallel \overline{a}/a$, respectively. These maximal A–contexts subsume all environments in which C takes part in response A. Indeed, one can check that $(C \parallel P_1) \Downarrow A$ and $(C \parallel P_2) \Downarrow A$.

For every $C \in \mathsf{PC}$ and $b \in \mathbb{B}$, we finally define

$$[\![C]\!]_2^b \quad =_{\mathrm{df}} \quad \{\langle A, I(A)\rangle \mid A \subseteq_{\mathrm{fin}} \Pi,\ \rho(C,A) = b,\ \text{ and } \langle A, I(A)\rangle \text{ is an } A\text{–context for } C\}.$$

The elements $\langle A, L\rangle \in [\![C]\!]_2^b$ are $(\cap, \subseteq)$ sub–semi–lattices L of 2^A that represent all the bounded context behaviors, i.e., environments, generating the joint response A. The superscript $b \in \mathbb{B}$ determines whether C is actively participating ($b = \mathit{tt}$) or only passively admitting ($b = \mathit{ff}$) the macro step resulting in A. In the latter case, the response must entirely come from the environment. This is reflected in the fact that all passive contexts $\langle A, L\rangle \in [\![C]\!]_2^{ff}$ are of the form $\langle A, L\rangle = \langle A, \{A\}\rangle$, which we will abbreviate as id_A for convenience. The passive A–context id_A means that the environment P must be equivalent to transition $\cdot/A$ in order for $(C \parallel P) \Downarrow A$ to hold. Another structural property, which we may take advantage of, is that an A–context $\mathcal{P}$ is contained in $[\![C]\!]_2^{tt}$ if and only if there exists a maximal A–context $\mathcal{P}_{\max} \in [\![C]\!]_2^{tt}$ with $\mathcal{P} \subseteq \mathcal{P}_{\max}$. Consequently, we only need to list the maximal elements of $[\![C]\!]_2^{tt}$ relative to any given response A. We now obtain our main theorem as a corollary to Prop. 4.6 and Thm. 3.5.

THEOREM 4.8 (Full Abstraction). *Let $C, D \in \mathcal{C}$. Then, $C \simeq D$ if and only if*

$$\bigcup_{i \in ind(C)} [\![C_i]\!]_2^{tt} = \bigcup_{j \in ind(D)} [\![D_j]\!]_2^{tt} \quad \textit{and} \quad \bigcap_{i \in ind(C)} [\![C_i]\!]_2^{ff} = \bigcap_{j \in ind(D)} [\![D_j]\!]_2^{ff}.$$

The proof of this theorem requires the following distributivity property stated in terms of admissible sets of transitions, which is proved in App. A.

LEMMA 4.9 (Distributivity). *Let $S, C, D \in \mathcal{C}$ be configurations, $E \subseteq_{fin} \Pi$, and $T \subset \mathcal{T}$. Then, T is E–admissible for $S \parallel (C + D)$ if and only if one of the following conditions holds:*

1. *$T \cap$ trans$(C) \neq \emptyset$, and T is E–admissible for $S \parallel C$.*
2. *$T \cap$ trans$(D) \neq \emptyset$, and T is E–admissible for $S \parallel D$.*
3. *$T \subseteq$ trans(S), and T is E–admissible for both $S \parallel C$ and $S \parallel D$.*

Moreover, in Case (1) we have $T \subseteq$ trans$(S \parallel C)$ and in Case (2) $T \subseteq$ trans$(S \parallel D)$.

Using this lemma, we are now going to prove Thm. 4.8.

Proof. [Theorem 4.8] We begin with two observations about A–contexts, for $A \subseteq_{\text{fin}} \Pi$. First, for every $P \in \mathsf{PC}$, consider the pair $Beh(P, A) =_{\text{df}} \langle A, L\rangle$ with $L =_{\text{df}} \{V(0) \mid (n, V) \in SM(P) \text{ and } V(n-1) = A\}$. For every $C \in \mathsf{PC}$, it possesses the property

$$(C \parallel P) \Downarrow A \text{ if and only if } Beh(P, A) \text{ is an } A\text{–context of } C. \tag{4.5}$$

This follows essentially from Thm. 3.4 and Def. 4.7 of A–contexts. Note that if A is not a classic model of P then $Beh(P, A)$ is not even a behavior. Second, suppose $\langle A, L\rangle$ is a $(\cap, \subseteq)$ sub-semi–lattice of 2^A. Then, by Thm. 3.11, there must exist a parallel configuration $P \in \mathsf{PC}$ in the events A and not using $\perp$, such that $Beh(P)$, when restricted to the events A, is identical to $\langle A, I(A)\rangle$, as well as $I(A) = L$. These also satisfy, for every $C \in \mathsf{PC}$, the property

$$(C \parallel P) \Downarrow A \text{ if and only if } \langle A, L\rangle \text{ is an } A\text{–context of } C. \tag{4.6}$$

Thm. 4.8 is now a consequence of Prop. 4.6 and the following facts. For all $A \subseteq_{\text{fin}} \Pi$, $D \in \mathsf{PC}$, and $b \in \mathbb{B}$:

$$\forall L\, \exists P.\ \langle A, L\rangle \in [\![D]\!]_2^b \text{ if and only if } \langle A, P\rangle \in [\![D]\!]_1^b, \text{ and} \tag{4.7}$$

$$\forall P\, \exists L.\ \langle A, P\rangle \in [\![D]\!]_1^b \text{ if and only if } \langle A, L\rangle \in [\![D]\!]_2^b. \tag{4.8}$$

For establishing Statements (4.7) and (4.8), we use Statements (4.6) and (4.5), respectively, together with the construction of behavior $Beh(P, A)$ and Thm. 3.5. In both cases, we also exploit that the triggering indicator ρ only depends on A, but not on the above P or L. Thm. 4.8 is derived from Statements (4.7) and (4.8) and from Prop. 4.6 in the obvious fashion. □

Let us consider some examples. For the configuration in Fig. 4.1, we have $[\![C]\!]_2^{tt} = \{\mathcal{P}_1, \mathcal{P}_2\}$ and $[\![C]\!]_2^{ff} = \emptyset$, where $\mathcal{P}_1$ and $\mathcal{P}_2$ are given in Equation (4.4). Note, that here and in the following, we only list maximal contexts. This structure can also be generated from the sum $D_1 + D_2$, where $D_1 =_{\mathrm{df}} ac/b \parallel \overline{b}/b \parallel \overline{c}/c$ and $D_2 =_{\mathrm{df}} bc/a \parallel \overline{b}/b \parallel \overline{a}/a$. One obtains

$$[\![D_1]\!]_2^{tt} = \{\, \mathcal{P}_1 \,\}, \qquad [\![D_2]\!]_2^{tt} = \{\, \mathcal{P}_2 \,\}, \qquad [\![D_1]\!]_2^{ff} = \{id_{\{b,c\}}\}, \qquad \text{and} \qquad [\![D_2]\!]_2^{ff} = \{id_{\{a,b\}}\}.$$

Hence, $[\![D_1]\!]_2^{tt} \cup [\![D_2]\!]_2^{tt} = [\![C]\!]_2^{tt}$ and $[\![D_1]\!]_2^{ff} \cap [\![D_2]\!]_2^{ff} = \emptyset = [\![C]\!]_2^{ff}$. By Thm. 4.8, then, $C \simeq D_1 + D_2$. The Statecharts axiom hidden in this example, which reflects a causality principle, is

$$a/b \parallel b/a \;\simeq\; a/b + b/a\,, \tag{4.9}$$

for any events $a, b \in \Pi$. Intuitively, this congruence states that, if a and b mutually depend on each other (left–hand side), then *either* a causes b *or* b causes a (right–hand side). We might call this the "*tie–break axiom*" or "*causality axiom.*" More specifically, we obtain the following semantics:

$$\begin{aligned}
[\![a/b \parallel b/a]\!]_2^{tt} &= \{\, \langle \{a,b\}, \{\{a\}\,,\, \{a,b\}\} \rangle\,,\, \langle \{a,b\}, \{\{b\}\,,\, \{a,b\}\} \rangle \,\} \\
[\![a/b \parallel b/a]\!]_2^{ff} &= \{\, id_\emptyset \,\} \\
[\![a/b]\!]_2^{tt} &= \{\, \langle \{a,b\}, \{\{a\}\,,\, \{a,b\}\} \rangle \,\} \\
[\![a/b]\!]_2^{ff} &= \{\, id_\emptyset\,,\, id_{\{b\}} \,\} \\
[\![b/a]\!]_2^{tt} &= \{\, \langle \{a,b\}, \{\{b\}\,,\, \{a,b\}\} \rangle \,\} \\
[\![b/a]\!]_2^{ff} &= \{\, id_\emptyset\,,\, id_{\{a\}} \,\}\,.
\end{aligned}$$

From this we compute $[\![a/b \parallel b/a]\!]_2^{tt} = [\![a/b]\!]_2^{tt} \cup [\![b/a]\!]_2^{tt}$ and $[\![a/b + b/a]\!]_2^{ff} = [\![a/b]\!]_2^{ff} \cap [\![b/a]\!]_2^{ff}$. Hence, with Thm. 4.8, the congruence in Equation (4.9) is obtained.

To finish off, we return to Sec. 3 and re–visit the compositionality problem in the light of our semantics. First of all, one verifies that $C_{79} = \overline{b}/a + b/a \cong \overline{b}/a \parallel b/a = C'_{79}$, as stated in Sec. 3. The semantics of parallel configuration $C'_{79} \in \mathsf{PC}$ is

$$[\![C'_{79}]\!]_2^{tt} \;=\; \{\, \langle\, \{a\}\,,\, \{\emptyset, \{a\}\} \,\rangle\,,\, \langle\, \{a,b\}\,,\, \{\{b\}, \{a,b\}\} \,\rangle \,\} \qquad \text{and} \qquad [\![C'_{79}]\!]_2^{ff} \;=\; \emptyset\,.$$

The active and passive contexts of b/a have been given above; it remains to analyze $\overline{b}/a$:

$$[\![\,\overline{b}/a\,]\!]_2^{tt} \;=\; \{\, \langle\, \{a\}\,,\, \{\emptyset, \{a\}\} \,\rangle \,\} \qquad \text{and} \qquad [\![\,\overline{b}/a\,]\!]_2^{ff} \;=\; \{\, id_{\{b\}}, id_{\{a,b\}} \,\}\,.$$

When combining the pieces, we obtain $[\![C'_{79}]\!]_2^{tt} = [\![\,\overline{b}/a\,]\!]_2^{tt} \cup [\![\,b/a\,]\!]_2^{tt}$ and $[\![C'_{79}]\!]_2^{ff} = \emptyset = [\![\,\overline{b}/a\,]\!]_2^{ff} \cap [\![\,b/a\,]\!]_2^{ff}$, whence $C'_{79} \simeq \overline{b}/a + b/a = C_{79}$. In addition, our semantics shows why configurations C_{79} and C'_{79} are distinguished from $C_{14} = \cdot/a \parallel b/a$. Configuration C_{14} has the active $\{a,b\}$–context $\langle \{a,b\}, \{\emptyset, \{b\}, \{a,b\}\} \rangle \in [\![C_{14}]\!]_2^{tt}$ which is not contained in $[\![C'_{79}]\!]_2^{tt}$ or in $[\![\,\overline{b}/a\,]\!]_2^{tt} \cup [\![\,b/a\,]\!]_2^{tt}$. This $\{a,b\}$–context $\langle \{a,b\}, \{\emptyset, \{b\}, \{a,b\}\} \rangle$ corresponds to context $\Phi_{56}[x] = x \parallel a/b$ used in Sec. 3 to differentiate C_{79} from C_{14}. It shows that $\Phi[C_{14}] \Downarrow \{a,b\}$ but $\Phi[C_{79}] \not\Downarrow \{a,b\}$.

With Thm. 4.8 (full abstraction) we have finally achieved our goal. Summarizing, the fully–abstract semantics developed in this report consists of the mapping $[\![\cdot]\!]_3$ given by

$$[\![C]\!]_3 =_{\mathrm{df}} \langle \;\; \bigcup_{i \in \mathsf{ind}(C)} [\![C_i]\!]_2^{tt} \;\;,\;\; \bigcap_{i \in \mathsf{ind}(C)} [\![C_i]\!]_2^{ff} \;\; \rangle\,.$$

Thm. 4.8 implies that $C \simeq D$ if and only if $[\![C]\!]_3 = [\![D]\!]_3$. This means that $[\![\cdot]\!]_3$ is compositional in the algebraic sense, i.e., if $[\![C]\!]_3 = [\![D]\!]_3$ then $[\![\Phi[C]]\!]_3 = [\![\Phi[D]]\!]_3$, for all contexts $\Phi[x]$. In contrast to $[\![C]\!]_1$, and indeed to the starting point $[\![C]\!]_0$, this fully–abstract interpretation $[\![C]\!]_3$ is both satisfactorily *semantic* and *finite*. It is also natural in that it realizes the obvious logical interpretation of (parallel) configurations as sequences of micro steps. Hence, the Statecharts semantics of Pnueli and Shalev is quite natural and elegant. Moreover, we believe that $[\![C]\!]_3$, in combination with Lemma 4.4 (expansion), directly lends itself to be applied for a model–based implementation of Pnueli and Shalev's semantics, which does not require backtracking for handling failure.

However, our semantics $[\![\cdot]\!]_3$ is not denotational, which would require that $[\![\Phi[C]]\!]_3$ is obtained directly from $[\![C]\!]_3$, when reading the syntactic operators of $\Phi[x]$ as suitable constructions in the semantic domain. As presented, the definition of $[\![C]\!]_3$ depends on the transformation of C into a sum form $\sum_{i \in \mathsf{ind}(C)} C_i$, which is a purely syntactic process. For a denotational semantics, this "normalization" would have to be performed directly in the semantic domain.

4.4. Conservativity. This section establishes that our extension of the standard Statecharts syntax by arbitrary choices $C + D$, where $D \in \mathsf{C}$, and by the failure event $\bot$ is conservative, i.e., the full–abstraction result regarding our configuration algebra is also true for the original Statecharts language. As a byproduct of our investigation, we obtain a proof for Prop. 4.1, too.

Formally, let C_f be some distinguished subset of C, and let PC_f be the parallel configurations in C_f, i.e., $\mathsf{PC}_f =_{\mathrm{df}} \mathsf{C}_f \cap \mathsf{PC}$. In the fragments C_f and PC_f, we consider two congruences $\simeq_f$ and $\simeq_f^+$, respectively, which are defined as follows:

$$C \simeq_f D \text{ if and only if } \forall \Phi[x] \in \mathsf{C}_f,\, E, A \subseteq_{\mathrm{fin}} \Pi.\;\; \Phi[C] \Downarrow_E A \text{ if and only if } \Phi[D] \Downarrow_E A\,.$$

$$C \simeq_f^+ D \text{ if and only if } \forall P \in \mathsf{PC}_f,\, E, A \subseteq_{\mathrm{fin}} \Pi,\, b \in \mathbb{B}.$$
$$((C \parallel P) \Downarrow_E A \text{ and } \rho(C, A) = b) \text{ if and only if } ((D \parallel P) \Downarrow_E A \text{ and } \rho(D, A) = b)\,.$$

In the special case $\mathsf{C}_f = \mathsf{C}$, we simply write $\simeq$ and $\simeq^+$ instead of $\simeq_f$ and $\simeq_f^+$, respectively. The key step towards our conservativity result is to show that, when fragment C_f encompasses a minimum amount of discriminating contexts, the equivalence between $C \simeq^+ D$ and $C \simeq_f^+ D$ entails the equivalence between $C \simeq_f D$ and $C \simeq_f^+ D$.

LEMMA 4.10. *Let C_f be a fragment of C satisfying the following two conditions: (i) C_f is closed under the operations $[\cdot] + t$ and $[\cdot] \parallel t$, for all transitions t in C_f, is closed under sub–configurations, and contains at least the transitions $\cdot/A$, for all $A \subseteq_{fin} \Pi \setminus \{\bot\}$; and (ii) $C \simeq^+ D$ if and only if $C \simeq_f^+ D$. Then, $C \simeq_f D$ if and only if $C \simeq_f^+ D$.*

The proof of this lemma can be found in App. A. A direct consequence of it, for the fragment $\mathsf{C}_f =_{\mathrm{df}} \mathsf{C}$, is Prop. 4.1 which essentially states that $C \simeq D$ is equivalent to $C \simeq^+ D$. As another consequence, consider the *standard* fragment $\mathsf{C}_s \subseteq \mathsf{C}$ of Statecharts, which consists of all configurations that (1) use the hierarchy operator only in the special form $[\cdot] + t$, for arbitrary transitions $t \in \mathcal{T}$, and (2) do not contain the failure event $\bot$ or its negation in any transition trigger or action.

Given an arbitrary parallel configuration $P \in \mathsf{PC}$, we define its *standardization* to be the configuration $P_s \in \mathsf{PC}_s$, obtained from P by dropping all transitions containing $\bot$ in their triggers or actions, as well as dropping all occurrences of $\overline{\bot}$ from the triggers of the remaining transitions. Note that P_s may be the empty configuration even though P is not. Obviously, by removing from P transitions with $\bot$ in their actions, we lose information about the failure behavior of P. In fact, P_s does not produce any failure due to the presence of events. For example, parallel configuration P might contain transition $a/\bot$. Then, P

produces a failure whenever the environment offers event a, but P_s does not since $a/\bot$ is dropped. To recover this information, we define, for every $P \in \mathsf{PC}$, a set $\mathit{fail}(P) \subseteq 2^{\Pi \setminus \{\bot\}}$ of those environments that would trigger a transition having $\bot$ in its actions and, hence, would produce a failure. More precisely, let $P_\bot \in \mathsf{PC}$ be the parallel composition of all transitions of P that have $\bot$ in their action. Then, $\mathit{fail}(P) =_{\mathrm{df}} \{A \subseteq_{\mathrm{fin}} \Pi \setminus \{\bot\} \,|\, \rho(P_\bot, A) = \mathit{tt}\}$. Taking into account the sets $\mathit{fail}(P)$, we can show that this standardization does not change the communication behavior of parallel configurations.

Lemma 4.11. *Let $C \in \mathsf{C}$, $A \subseteq_{\mathit{fin}} \Pi$, and $P \in \mathsf{PC}$. Then, $(C \,\|\, P) \Downarrow A$ if and only if $(C \,\|\, P_s) \Downarrow A$ and $A \notin \mathit{fail}(P)$.*

Proof. Let $P \in \mathsf{PC}$ and $A \subseteq_{\mathrm{fin}} \Pi \setminus \{\bot\}$ be arbitrary. We first prove that

$$A \notin \mathit{fail}(P) \text{ implies } \mathsf{triggered}(P, A) = \mathsf{triggered}(P_s, A)\,. \tag{4.10}$$

Inclusion $\mathsf{triggered}(P_s, A) \subseteq \mathsf{triggered}(P, A)$ is trivial since the transitions of P_s are a subset of those of P, possibly having an extra trigger event $\overline{\bot}$ in P, which does not affect their enabling as $\bot \notin A$. For the inclusion $\mathsf{triggered}(P, A) \subseteq \mathsf{triggered}(P_s, A)$, we assume $A \notin \mathit{fail}(P)$. Let $t \in \mathsf{triggered}(P, A)$, i.e., t is a transition of P enabled by A. Since $A \notin \mathit{fail}(P)$, transition t does not have event $\bot$ in its action. Similarly, it cannot have $\bot$ in its trigger; otherwise, it would not be enabled, given $\bot \notin A$. This means that t must be contained in P_s, with any $\overline{\bot}$ in its trigger removed. In any case, transition t is still enabled. Hence, $\mathsf{triggered}(P, A) \subseteq \mathsf{triggered}(P_s, A)$. Statement (4.10) implies that $(C \,\|\, P) \Downarrow A$ if and only if $(C \,\|\, P_s) \Downarrow A$ and $A \notin \mathit{fail}(P)$, for all $C \in \mathsf{C}$, since P is a parallel context, and that $(D \,\|\, R) \Downarrow A$ implies $\bot \notin A$ and $A \notin \mathit{fail}(R)$, for any configurations $D \in \mathsf{C}$ and $R \in \mathsf{PC}$. □

As a consequence of the above lemma, we now obtain the desired result for the standard fragment.

Lemma 4.12. *Let $C, D \in \mathsf{C}$. Then, $C \simeq^+ D$ if and only if $C \simeq^+_s D$.*

Proof. Direction "$\Longrightarrow$" is trivial since the standard parallel contexts are just a special class of parallel contexts. For the other direction, suppose $C \simeq^+_s D$. Let $P \in \mathsf{PC}$, $A \subseteq_{\mathrm{fin}} \Pi$, and $b \in \mathbb{B}$ be such that $(C \,\|\, P) \Downarrow A$ and $\rho(C, A) = b$. By direction "$\Longrightarrow$" of Lemma 4.11, $(C \,\|\, P_s) \Downarrow A$ and $A \notin \mathit{fail}(P)$. Since $P_s \in \mathsf{PC}_s$ and $C \simeq^+_s D$ we infer $(D \,\|\, P_s) \Downarrow A$ and $\rho(D, A) = b$. Another application of Lemma 4.11, this time direction "$\Longleftarrow$" for configuration D, yields $(D \,\|\, P) \Downarrow A$. Hence, we have shown that, for all $P \in \mathsf{PC}$, $A \subseteq_{\mathrm{fin}} \Pi$, and $b \in \mathbb{B}$,

$$((C \,\|\, P) \Downarrow A \text{ and } \rho(C, A) = b) \;\text{ implies }\; ((D \,\|\, P) \Downarrow A \text{ and } \rho(D, A) = b)\,.$$

Since our argument is symmetric in C and D, we can establish the other direction, too. □

We are now ready to summarize the conservativity properties.

Theorem 4.13 (Conservativity). *For arbitrary $C, D \in \mathsf{C}$, the following statements are equivalent:*

$$(1)\; C \simeq D\,, \qquad (2)\; C \simeq^+ D\,, \qquad (3)\; C \simeq_s D\,, \quad \text{and} \qquad (4)\; C \simeq^+_s D\,.$$

Proof. The equivalence "(1) $\Longleftrightarrow$ (2)" follows from Lemma 4.10 for the fragment $\mathsf{C}_f =_{\mathrm{df}} \mathsf{C}$, whereas equivalence "(2) $\Longleftrightarrow$ (4)" is the statement of Lemma 4.12. Finally, equivalence "(3) $\Longleftrightarrow$ (4)" arises from specializing Lemma 4.10 to fragment C_s, using result "(2) $\Longleftrightarrow$ (4)" and the fact that C_s satisfies Assumption (i) required in Lemma 4.10. □

The equivalence of $C \simeq D$ and $C \simeq_s D$ is a crucial result since it shows that there are no additional semantic distinctions introduced by our use of a more general configuration syntax. Hence, whenever we restrict ourselves to the standard fragment we obtain exactly the same compositional semantics as if we had used the restricted language in the first place. This substantiates our claim that our semantics is fully abstract for Statecharts and the operational step semantics of Pnueli and Shalev, despite the fact that we are employing a slightly richer syntax.

5. Related Work. Our investigation focused on Pnueli and Shalev's original presentation [17] of Statecharts and its macro–step semantics. Like [17] we only consider single macro steps since it is here where the main challenge for a fully–abstract semantics of Statecharts lies. The elegance of Pnueli and Shalev's operational semantics manifests itself in the existence of an equivalent *declarative fixed point semantics.* However, as illustrated in [17], this equivalence breaks down when allowing disjunctions in transition triggers. For example, the configurations $(\overline{a} \vee b)/a$ and $\overline{a}/a \;\|\; b/a$ do not have, as was expected, the same response behavior. This subtlety can be explained in our intuitionistic framework. In Pnueli and Shalev's setting, $\overline{a} \vee b$ is classically interpreted as "*throughout the macro step, not a or b.*" In contrast, this report's approach reads the configuration as "*throughout the macro step not a, or throughout the macro step b.*" Our stronger intuitionistic interpretation restores the coincidence of operational and declarative semantics. This assumes, of course, that the former is adjusted accordingly, which is not difficult, however. The step procedure must only ensure that, whenever transition $(\overline{a} \vee b)/a$ is fired due to absence of a, event a is prohibited to occur in any subsequent micro step. Our approach also suggests other extensions to larger fragments of intuitionistic logic, such as "higher–order" transitions, e.g., $(a \supset b) \supset c$, which may be explored in the future.

Our framework can also be employed for analyzing various other asynchronous Statecharts variants with global consistency. One example is the work of Maggiolo–Schettini et al. [15], which is inspired by the process–algebraic semantics presented in [13, 18]. In [15], and also in [14], the step–construction procedure cannot fail since a transition is only considered to be enabled, if it is enabled in the sense of Pnueli and Shalev *and* if it does not produce any event that violates global consistency. This novel semantics is specified using a notion of *compatibility* [15] which introduces a look–ahead concept for avoiding failures during the construction of macro steps. As an example, consider configuration $C =_{\mathrm{df}} t_1 \;\|\; t_2$, where $t_1 =_{\mathrm{df}} a/b$ and $t_2 =_{\mathrm{df}} \overline{b}/a$. According to [15], when C is evaluated in the empty environment, the response $\{a\}$ is obtained: First, transition t_2 fires due to the absence of event b, thereby producing event a. The presence of a now satisfies the trigger of t_1. Its execution would introduce event b, whence transition t_1 is incompatible with t_2 which has fired due to the absence of event b. Therefore, transition t_1 is disabled in [15]. In Pnueli and Shalev's original semantics, however, t_1 is enabled with the consequence that the step construction is forced to fail. The difference between the two semantics can be explained in terms of stabilization sequences. While Pnueli and Shalev take t_1 to stand for the specification $a \supset b$ and t_2 for $\neg b \supset a$, Maggiolo–Schettini et al. apply the interpretation $a \supset (b \vee \neg b)$ for t_1 and $\neg b \supset (a \vee \neg a)$ for t_2. Thus, e.g., t_1 is read as "*if a becomes present then either b is asserted or b never becomes present.*" The second case "*b never becomes present*" accommodates the possibility that t_1, even though its trigger a is satisfied, is not taken due to an incompatibility with another transition in the environment that requires the global absence of b. A similar remark applies to transition t_2. Indeed, one can show that configuration

$$C_{enc} =_{\mathrm{df}} t_1 \,\|\, t_2 \;=\; (a \supset (b \vee \neg b)) \;\wedge\; (\neg b \supset (a \vee \neg a))$$

possesses $\{a\}$ as a response model, in the sense of Def. 3.3, which is in accordance with the operational semantics of [15]. Note that this encoding, again, crucially depends on the fact that $a \vee \neg a$ differs from *true* in intuitionistic logic. Generalizing this example, we conjecture that the transition semantics of [14, 15] can be captured in terms of response models by reading a transition E/A as formula $E \supset (A \vee \neg A)$. Of course, our language of configurations needs to be extended to allow disjunctions as part of transition actions. We further want to remark that it is possible to translate between the two considered semantics [15, 17] using our framework. For instance, the sequence model semantics of C_{enc} may be captured by configuration $a/b + \bar{b}/a$. This configuration has the same operational behavior in Pnueli and Shalev's step semantics as C has in [15]. Moreover, we expect that our semantics may also be useful to derive full–abstractness results for the semantics in [15] and other Statecharts semantics with global consistency. Especially, lifting our results to sequences of macro steps should not present any major difficulties when employing the standard framework of transition systems.

Other investigations into the compositionality problem of Statecharts were conducted by Uselton and Smolka [18] who model Statecharts' macro steps by labeled transition systems in a process–algebraic style. They achieve compositionality by using partial orders on events, which encode causality information, as transition labels. As was pointed out by Levi in [13], the partial orders on events used by Uselton and Smolka are not sufficient to capture Pnueli and Shalev's semantics faithfully. Levi's semantics remedies the problem by employing partial orders on *sets* of events. Although this semantics complies with the one of Pnueli and Shalev, no full–abstraction result is presented. It should be noted that our semantics, too, uses a lattice–theoretic structure on sets of events. The elements $\langle A, L\rangle$ of $[\![C]\!]_2^{tt}$, which represent the active responses of C, are $(\cap, \subseteq)$ sub–lattices of 2^A that correspond to the transition labels in Levi's work. The main difference between our approach and the ones in [13, 18] is that our lattices do not contain any negative events, whence they may be considered more semantic in nature. The precise relationship between our semantics and that of [13] still needs to be explored.

Our intuitionistic approach is also related to recent work in *synchronous languages*, especially Berry's ESTEREL [3]. In ESTEREL, causality is traditionally treated separately from compositionality and synchrony, as part of type–checking specifications. If the (conservative) type checker finds causality to be violated, it rejects the specification under consideration. Otherwise, the specification's semantics can be determined in a very simple fashion, since one may — in contrast to Statecharts semantics — abstract from the construction details of macro steps while preserving compositionality. This was shown by Broy in [5], using a domain–theoretic account of abstracting from a sequence of micro steps to a macro step based on *streams*. The more recent Version 5 of ESTEREL, however, replaces the restrictive treatment of causality by defining a semantics via a particular Boolean logic that is *constructive* [2], as is intuitionistic logics. The constructive semantics of ESTEREL is especially interesting since it relates to the traditional semantics for digital circuits [2, 4].

Denotational semantics and full abstraction were also studied by Huizing et al. [10, 11] for an early and lateron rejected Statecharts semantics [9]. In particular, that semantics does not consider global consistency, which makes their result largely incomparable to ours. Also, the abstractness result is proved with respect to a richer set of syntactic operators than we consider here. Finally, it should be mentioned that the lack of compositionality of Statecharts semantics inspired the development of new visual languages, such as Alur et al.'s *communicating hierarchical state machines* [1], Maraninchi's ARGOS [16], and Leveson's RSML [12].

6. Conclusions and Future Work. To the best of our knowledge, this is the first report to present a fully–abstract Statecharts semantics for Pnueli and Shalev's original macro–step semantics [17]. The latter semantics was found to be non–compositional as it employs classic logic for interpreting macro steps. In contrast, our semantics borrows ideas from intuitionistic logic. It encodes macro steps via stabilization sequences which we characterized using semi–lattice structures, called behaviors. Behaviors capture the interactions between Statecharts and their environments and consistently combine the notions of causality, global consistency, and synchrony in a model–theoretic fashion. Thus, our approach suggests a model–based implementation of Pnueli and Shalev's semantics, thereby eliminating the need to implement failure via backtracking. It further permits the introduction of more general trigger conditions, including disjunctions, which solves some of the difficulties reported in [17].

Regarding future work, several further theoretical investigations need to be conducted. First, we plan to derive a fully–abstract *denotational* semantics for Statecharts on the basis of our results. To this end, we need to find a semantic mapping that does not depend on a syntactic normalization. Second, the macro–step semantics for single configurations should be lifted to the full Statecharts semantics which involves sequences of macro steps. We also intend to employ our framework for developing algebraic characterizations of step congruence and for uniformly comparing various variants of Statecharts' macro–step semantics studied in the literature [13, 14, 15]. Practical applications of our work include semantic–based program transformations, abstract analyses, and compositional code generation.

Acknowledgments. We would like to thank the members of the Sheffield Verification and Testing Group as well as the anonymous referees of a conference version of this report for their valuable comments and suggestions.

REFERENCES

[1] R. Alur, S. Kannan, and M. Yannakakis, *Communicating hierarchical state machines*, in 26th International Colloquium on Automata, Languages and Programming (ICALP '99), P. van Emde Boas, J. Wiedermann, and M. Nielsen, eds., Vol. 1644 of Lecture Notes in Computer Science, Prague, Czech Republic, July 1999, Springer-Verlag, pp. 169–178.

[2] G. Berry, *The constructive semantics of pure Esterel.* Draft Version 3. Available at http://www-sop.inria.fr/meije/Personnel/Gerard.Berry.html, 1999.

[3] G. Berry and G. Gonthier, *The Esterel synchronous programming language: Design, semantics, implementation*, Science of Computer Programming, 19 (1992), pp. 87–152.

[4] G. Berry and E. Sentovich, *An implemenatation of constructive synchronous programs in POLIS.* Available at http://www-sop.inria.fr/meije/Personnel/Gerard.Berry.html, 1999.

[5] M. Broy, *Abstract semantics of synchronous languages: The example Esterel*, Tech. Report TUM-I9706, Munich Univ. of Technology, Germany, March 1997.

[6] W. Damm, B. Josko, H. Hungar, and A. Pnueli, *A compositional real-time semantics of STATEMATE designs*, in Compositionality: The Significant Difference, W. de Roever, H. Langmaack, and A. Pnueli, eds., Vol. 1536 of Lecture Notes in Computer Science, Bad Malente, Germany, September 1997, Springer-Verlag, pp. 186–238.

[7] D. Harel, *Statecharts: A visual formalism for complex systems*, Science of Computer Programming, 8 (1987), pp. 231–274.

[8] D. Harel and A. Naamad, *The STATEMATE semantics of Statecharts*, ACM Transactions on Software Engineering, 5 (1996), pp. 293–333.

[9] D. Harel, A. Pnueli, J. Pruzan-Schmidt, and R. Sherman, *On the formal semantics of Statecharts*, in Symposium on Logic in Computer Science (LICS '87), Ithaca, NY, USA, June 1987, IEEE Computer Society Press, pp. 56–64.

[10] C. Huizing, *Semantics of Reactive Systems: Comparison and Full Abstraction*, Ph.D. thesis, Eindhoven University of Technology, The Netherlands, March 1991.

[11] C. Huizing, R. Gerth, and W. de Roever, *Modeling Statecharts behavior in a fully abstract way*, in 13th Colloquium on Trees and Algebra in Programming (CAAP '88), M. Dauchet and M. Nivat, eds., Vol. 299 of Lecture Notes in Computer Science, Nancy, France, March 1988, Springer-Verlag, pp. 271–294.

[12] N. Leveson, M. Heimdahl, H. Hildreth, and J. Reese, *Requirements specification for process-control systems*, IEEE Transactions on Software Engineering, 20 (1994).

[13] F. Levi, *Verification of Temporal and Real-time Properties of Statecharts*, Ph.D. thesis, University of Pisa-Genova-Udine, Italy, February 1997.

[14] G. Lüttgen, M. von der Beeck, and R. Cleaveland, *Statecharts via process algebra*, in 10th International Conference on Concurrency Theory (CONCUR '99), J. Baeten and S. Mauw, eds., Vol. 1664 of Lecture Notes in Computer Science, Eindhoven, The Netherlands, August 1999, Springer-Verlag, pp. 399–414.

[15] A. Maggiolo-Schettini, A. Peron, and S. Tini, *Equivalences of Statecharts*, in 7th International Conference on Concurrency Theory (CONCUR '96), U. Montanari and V. Sassone, eds., Vol. 1119 of Lecture Notes in Computer Science, Pisa, Italy, August 1996, Springer-Verlag, pp. 687–702.

[16] F. Maraninchi, *Operational and compositional semantics of synchronous automaton compositions*, in 3rd International Conference on Concurrency Theory (CONCUR '92), R. Cleaveland, ed., Vol. 630 of Lecture Notes in Computer Science, Stony Brook, NY, USA, August 1992, Springer-Verlag, pp. 550–564.

[17] A. Pnueli and M. Shalev, *What is in a step: On the semantics of Statecharts*, in Theoretical Aspects of Computer Software (TACS '91), T. Ito and A. Meyer, eds., Vol. 526 of Lecture Notes in Computer Science, Sendai, Japan, September 1991, Springer-Verlag, pp. 244–264.

[18] A. Uselton and S. Smolka, *A compositional semantics for Statecharts using labeled transition systems*, in 5th International Conference on Concurrency Theory (CONCUR '94), B. Jonsson and J. Parrow, eds., Vol. 836 of Lecture Notes in Computer Science, Uppsala, Sweden, August 1994, Springer-Verlag, pp. 2–17.

[19] D. van Dalen, *Intuitionistic logic*, in Handbook of Philosophical Logic, Vol. III, Reidel, 1986, ch. 4, pp. 225–339.

[20] M. von der Beeck, *A comparison of Statecharts variants*, in 3rd International School and Symposium on Formal Techniques in Real-time and Fault-tolerant Systems (FTRTFT '94), H. Langmaack, W. de Roever, and J. Vytopil, eds., Vol. 863 of Lecture Notes in Computer Science, Lübeck, Germany, September 1994, Springer-Verlag, pp. 128–148.

Appendix A. Proofs of Lemmas 4.4 and 4.10. We first prove Lemma 4.4.

Proof. First note that we may assume $T \subseteq \mathsf{trans}(C) \cup \mathsf{trans}(D) \cup \mathsf{trans}(S)$. Otherwise, T would not be admissible for any of $S \parallel (C + D)$, or $S \parallel C$, or $S \parallel D$, in which case the statement of the theorem would be trivially true. Under this assumption, then, Conds. (1) $T \cap \mathsf{trans}(C) \neq \emptyset$, (2) $T \cap \mathsf{trans}(D) \neq \emptyset$, and (3) $T \subseteq \mathsf{trans}(S)$ cover all possible cases. We first derive a few simple facts about the relationship between function $\mathsf{enabled}$ for $S \parallel (C + D)$, on the one hand, and $\mathsf{enabled}$ for $S \parallel C$ and $S \parallel D$, on the other hand. We start off by stating the equality

$$\mathsf{enabled}(S \parallel (C + D), E, T') \;=\; \mathsf{enabled}(S \parallel C, E, T') \cup \mathsf{enabled}(S \parallel D, E, T') \tag{A.1}$$

which can be proved by a straightforward calculation employing the definition of $\mathsf{enabled}$. Next, observe that, for all $T'' \subseteq \mathsf{trans}(S \parallel C)$ and all sets $T' \subseteq \mathcal{T}$ of transitions, we have

$$\mathsf{enabled}(S \parallel (C + D), E, T') \cap T'' \;=\; \mathsf{enabled}(S \parallel C, E, T') \cap T'' \tag{A.2}$$

and, symmetrically, if $T'' \subseteq \mathsf{trans}(S \parallel D)$, then

$$\mathsf{enabled}(S \parallel (C + D), E, T') \cap T'' \;=\; \mathsf{enabled}(S \parallel D, E, T') \cap T''\,. \tag{A.3}$$

The proofs of these statements are quite easy, as in the first case, we have $\mathsf{consistent}(S \parallel (C + D), T') \cap T'' = \mathsf{consistent}(S \parallel C, T') \cap T''$; in the second case, $\mathsf{consistent}(S \parallel (C + D), T') \cap T'' = \mathsf{consistent}(S \parallel D, T') \cap T''$. Now we proceed to prove our distributivity lemma. We begin with Case (1), i.e., $T \cap \mathsf{trans}(C) \neq \emptyset$. Suppose that T is E–admissible for $S \parallel (C + D)$. Since T contains at least one transition from C, it cannot include any transition from D; otherwise, T would not be consistent for $S \parallel (C + D)$. Thus, $\emptyset \neq T \subseteq \mathsf{trans}(S \parallel C)$. The property of E–admissibility and Prop. (A.2) imply

$$T \;=\; \mathsf{enabled}(S \parallel (C + D), E, T) \;=\; \mathsf{enabled}(S \parallel C, E, T) \cap \mathsf{trans}(S \parallel C) \;=\; \mathsf{enabled}(S \parallel C, E, T)\,. \tag{A.4}$$

Now let $T' \subset T$ be given. Since T is E–inseparable and $T \subseteq \mathsf{trans}(S \parallel C)$, we have by Prop. (A.2) that

$$\mathsf{enabled}(S \parallel C, E, T') \cap (T \setminus T') \;=\; \mathsf{enabled}(S \parallel (C + D), E, T') \cap (T \setminus T') \;\neq\; \emptyset\,.$$

Together with Prop. (A.4), this shows that T must be E–admissible for $S \parallel C$. Vice versa, if T is E–admissible for $S \parallel C$, then $\emptyset \neq T \subseteq \mathsf{trans}(S \parallel C)$, too. Again, Prop. (A.4) implies that T is E–admissible for $S \parallel (C + D)$. Case (2) is handled completely symmetrically to Case (1), using Prop. (A.3) to establish $T = \mathsf{enabled}(S \parallel (C + D), E, T) = \mathsf{enabled}(S \parallel D, E, T)$ as well as the E–inseparability of T for $S \parallel D$.

It remains to consider Case (3), i.e., $T \subseteq \mathsf{trans}(S)$; in particular, $T \subseteq \mathsf{trans}(S \parallel C)$. If T is E–admissible for $S \parallel (C + D)$, then $T = \mathsf{enabled}(S \parallel (C + D), E, T)$. We use Prop. (A.2) to obtain $\mathsf{enabled}(S \parallel C, E, T) = \mathsf{enabled}(S \parallel C, E, T) \cap T = \mathsf{enabled}(S \parallel (C + D), E, T) \cap T = T$. We can also employ Prop. (A.2) to show that T is inseparable for $S \parallel C$. If $T' \subset T$, then $(T \setminus T') \subseteq \mathsf{trans}(S)$; as a consequence, $\mathsf{enabled}(S \parallel C, E, T') \cap (T \setminus T') = \mathsf{enabled}(S \parallel (C + D), E, T') \cap (T \setminus T') \neq \emptyset$, where the inequality is due to the E–inseparability of T with respect to $S \parallel (C + D)$ and to the first equation derived from Prop. (A.2). This completes the proof that T is E–admissible for $S \parallel C$. In an analogous fashion, one can show that T is E–admissible for $S \parallel D$ using Prop. (A.3). For the other direction of Case (3), assume that $T \subseteq \mathsf{trans}(S)$ is E–admissible for both $S \parallel C$ and $S \parallel D$. From Prop. (A.1), for arbitrary $T' \subseteq T$, we conclude $\mathsf{enabled}(S \parallel (C + D), E, T') = \mathsf{enabled}(S \parallel C, E, T') \cup \mathsf{enabled}(S \parallel D, E, T')$. An immediate consequence of $\mathsf{enabled}(S \parallel C, E, T) = T = \mathsf{enabled}(S \parallel D, E, T)$ is that $\mathsf{enabled}(S \parallel (C + D), E, T) = T$. Moreover, for any

$T' \subset T$, and since $\mathsf{enabled}(S \parallel C, E, T') \cap (T \backslash T') \neq \emptyset$, we also have $\mathsf{enabled}(S \parallel (C + D), E, T') \cap (T \backslash T') \neq \emptyset$. Thus, T is E–admissible for $S \parallel (C + D)$. □

We are now able to establish Lemma 4.10.

Proof. [Lemma 4.10] For proving direction "$\Longleftarrow$", suppose that $C \simeq_f^+ D$. By Assumption (ii) this is equivalent to $C \simeq^+ D$, i.e., we have

$$((C \parallel P) \Downarrow_E A \text{ and } \rho(C, A) = b) \text{ if and only if } ((D \parallel P) \Downarrow_E A \text{ and } \rho(D, A) = b), \qquad \text{(A.5)}$$

for all parallel configurations $P \in \mathsf{PC}$, event sets $E, A \subseteq_{\mathsf{fin}} \Pi$, and $b \in \mathbb{B}$. We must show that

$$\Phi[C] \Downarrow_E A \text{ if and only if } \Phi[D] \Downarrow_E A, \qquad \text{(A.6)}$$

for all contexts $\Phi[x] \in \mathsf{C}_f$ and $E, A \subseteq_{\mathsf{fin}} \Pi$. We shall prove the following somewhat stronger invariant by induction on the structure of contexts $\Phi[x]$. For every configuration $S \in \mathsf{C}$ and every set T_1 of transitions such that T_1 is admissible for $S \parallel \Phi[C]$, there exists a set T_2 of transitions, which is admissible for $S \parallel \Phi[D]$, such that

$$\begin{aligned} \mathsf{act}(T_1) \quad &= \quad \mathsf{act}(T_2), && \text{(A.7)} \\ T_1 \cap \mathsf{trans}(S) \quad &= \quad T_2 \cap \mathsf{trans}(S), \text{ and} && \text{(A.8)} \\ T_1 \subseteq \mathsf{trans}(S) \text{ if and only if } & T_2 \subseteq \mathsf{trans}(S). && \text{(A.9)} \end{aligned}$$

Observing that any initial set E of environment events may be accounted for in configuration S, it immediately follows that $\Phi[C] \Downarrow_E A$ implies $\Phi[D] \Downarrow_E A$. Since the other direction is obtained by symmetry, the proof of direction "$\Longleftarrow$" of the proposition is then completed. In the following, we first deal with the induction step and subsequently with the slightly more complicated base case.

- Case $\Phi[x] = R \parallel \Psi[x]$ is a trivial application of the induction hypothesis which is phrased so that it quantifies over arbitrary parallel contexts. Note that $\Psi[x] \in \mathsf{C}_f$ by Prop. (i) for fragment C_f (closure under sub–configurations). Let T_1 be admissible for $S \parallel R \parallel \Psi[C]$. When taking $S' =_{\mathrm{df}} S \parallel R$ and applying the induction hypothesis to $S' \parallel \Psi[C]$, we obtain a set T_2 of transitions which is admissible for $S \parallel R \parallel \Psi[D]$, such that

  $$\begin{aligned} \mathsf{act}(T_1) \quad &= \quad \mathsf{act}(T_2), \\ T_1 \cap \mathsf{trans}(S \parallel R) \quad &= \quad T_2 \cap \mathsf{trans}(S \parallel R), \text{ and} \\ T_1 \subseteq \mathsf{trans}(S \parallel R) \text{ if and only if } & T_2 \subseteq \mathsf{trans}(S \parallel R). \end{aligned}$$

  The first equality yields Equation (A.7). Moreover, since $\mathsf{trans}(S \parallel R) = \mathsf{trans}(S) \cup \mathsf{trans}(R)$ and $\mathsf{trans}(S) \cap \mathsf{trans}(R) = \emptyset$, the last two equivalences imply $T_1 \cap \mathsf{trans}(S) = T_2 \cap \mathsf{trans}(S)$, as well as $T_1 \subseteq \mathsf{trans}(S)$ if and only if $T_2 \subseteq \mathsf{trans}(S)$, as required for Props. (A.8) and (A.9).
- When the context is of form $\Phi[x] = R + \Psi[x]$, for some $\Psi[x] \in \mathsf{C}_f$, we let T_1 be admissible for $S \parallel (R + \Psi[C])$. By Lemma 4.9 (distributivity) we have to consider three cases:

  1. $T_1 \cap \mathsf{trans}(R) \neq \emptyset$, and T is admissible for $S \parallel R$.
  2. $T_1 \cap \mathsf{trans}(\Psi[C]) \neq \emptyset$, and T_1 is admissible for $S \parallel \Psi[C]$.
  3. $T_1 \subseteq \mathsf{trans}(S)$, and T_1 is admissible for both $S \parallel R$ and $S \parallel \Psi[C]$.

  In Case (1), we immediately have that T_1 is admissible for $S \parallel (R + \Psi[D])$, simply by applying Lemma 4.9(1) backwards. Hence, we may choose $T_2 =_{\mathrm{df}} T_1$ to satisfy Equations (A.7)–(A.9). In

Cases (2) and (3), we appeal to the induction hypothesis as applied to context $\Psi[x]$. This yields a set T_2 of transitions, which is admissible for $S \parallel \Psi[D]$, with the properties

$$\begin{array}{rcl} \mathsf{act}(T_1) & = & \mathsf{act}(T_2)\,, \\ T_1 \cap \mathsf{trans}(S) & = & T_2 \cap \mathsf{trans}(S)\,, \text{ and} \\ T_1 \subseteq \mathsf{trans}(S) & \text{if and only if} & T_2 \subseteq \mathsf{trans}(S)\,. \end{array}$$

This proves Props. (A.7)–(A.9) for Cases (2) and (3) together. What remains to be seen is that T_2 is admissible for $S \parallel (R + \Psi[D])$. We demonstrate this separately for Cases (2) and (3).

Let Case (2) be given, i.e., $T_1 \cap \mathsf{trans}(\Psi[C]) \neq \emptyset$. This implies $T_1 \not\subseteq \mathsf{trans}(S)$, whence $T_2 \not\subseteq \mathsf{trans}(S)$ by Prop. (A.9). Since $T_2 \subseteq \mathsf{trans}(S \parallel \Psi[D])$ we obtain $T_2 \cap \mathsf{trans}(\Psi[D]) \neq \emptyset$. According to Lemma 4.9 (distributivity), T_2 is admissible for $S \parallel (R + \Psi[D])$, which was to be shown.

Finally, suppose we have Case (3), i.e., $T_1 \subseteq \mathsf{trans}(S)$, and T_1 is admissible for $S \parallel R$. By Prop. (A.9), $T_2 \subseteq \mathsf{trans}(S)$ and, further, $T_1 = T_1 \cap \mathsf{trans}(S) = T_2 \cap \mathsf{trans}(S) = T_2$ by Prop. (A.8). But then T_2 is admissible not only for $S \parallel \Psi[D]$ but also for $S \parallel R$. Hence, T_2 is admissible for $S \parallel (R + \Psi[D])$ by Lemma 4.9. This completes our consideration of context $\Phi[x] = R + \Psi[x]$.

- It remains to prove the base case $\Phi[x] = x$. Suppose T_1 is admissible for $S \parallel \Phi[C] = S \parallel C$, where $S \in \mathsf{C}$ is an arbitrary configuration. Let S_1 be the parallel composition of all transitions from S that are contained in T_1. We will use S_1 to refer both to this parallel configuration as well as to the subset of transition names of T_1, depending on the context. It is not difficult to show that T_1 is admissible for $S_1 \parallel C$, whence $(S_1 \parallel C) \Downarrow \mathsf{act}(T_1)$. By Assumption (A.5), then, $(S_1 \parallel D) \Downarrow \mathsf{act}(T_1)$, i.e., there must exist a set of transitions such that $T_2 \subseteq \mathsf{trans}(S_1 \parallel D) \subseteq \mathsf{trans}(S \parallel D)$ and such that T_2 is admissible for $S_1 \parallel D$ and

$$A =_{\mathrm{df}} \mathsf{act}(T_2) = \mathsf{act}(T_1)\,, \text{ and} \tag{A.10}$$

$$\mathsf{triggered}(C, A) = \emptyset \text{ if and only if } \mathsf{triggered}(D, A) = \emptyset\,. \tag{A.11}$$

Note, Prop. (A.11) is equivalent to $\rho(C, A) = \rho(D, A)$. From Props. (A.10) and (A.11) it follows, so we claim,

$$T_1 \cap \mathsf{trans}(S) \quad = \quad T_2 \cap \mathsf{trans}(S)\,, \text{ and} \tag{A.12}$$

$$T_1 \subseteq \mathsf{trans}(S) \text{ if and only if } T_2 \subseteq \mathsf{trans}(S)\,. \tag{A.13}$$

Because of $T_1 \subseteq \mathsf{trans}(S_1 \parallel C)$ and $T_2 \subseteq \mathsf{trans}(S_1 \parallel D)$, Prop. (A.12) is equivalent to $T_1 \cap \mathsf{trans}(S_1) = T_2 \cap \mathsf{trans}(S_1)$. The argument behind this is as follows. Since S_1 is a parallel composition of single transitions and since T_1 is admissible for $S_1 \parallel C$, set $T_1 \cap \mathsf{trans}(S_1)$ must consist precisely of all transitions of S_1 that are enabled by A. Hence, $T_1 \cap \mathsf{trans}(S_1) = \mathsf{triggered}(S_1, A)$. The same is true of T_2. Hence, $T_1 \cap \mathsf{trans}(S_1) = T_2 \cap \mathsf{trans}(S_1)$, as desired, which proves Prop. (A.12).

Prop. (A.13) is a consequence of Prop. (A.11). Since T_1 is admissible for $S_1 \parallel C$, we have $T_1 = \mathsf{consistent}(S_1 \parallel C, T_1) \cap \mathsf{triggered}(S_1 \parallel C, A)$. This implies that $T_1 \subseteq \mathsf{trans}(S_1)$ is equivalent to $\mathsf{triggered}(C, A) = \emptyset$. For if $T_1 \subseteq \mathsf{trans}(S_1)$, then $\mathsf{triggered}(C, A) \subseteq (\mathsf{trans}(S_1) \cup \mathsf{trans}(C)) \cap (\mathsf{triggered}(S_1, A) \cup \mathsf{triggered}(C, A)) = \mathsf{consistent}(S_1 \parallel C, T_1) \cap \mathsf{triggered}(S_1 \parallel C, A) = T_1 \subseteq \mathsf{trans}(S_1)$. This can only be true if $\mathsf{triggered}(C, A) = \emptyset$. Vice versa, suppose $T_1 \not\subseteq \mathsf{trans}(S_1)$. As $T_1 \subseteq \mathsf{trans}(S_1) \cup \mathsf{trans}(C)$, this means $T_1 \cap \mathsf{trans}(C) \neq \emptyset$. But then, because of $T_1 \cap \mathsf{trans}(C) = \mathsf{consistent}(S_1 \parallel C, T_1) \cap \mathsf{triggered}(S_1 \parallel C, A) \cap \mathsf{trans}(C) \subseteq \mathsf{triggered}(S_1 \parallel C, A) \cap \mathsf{trans}(C) = \mathsf{triggered}(C, A)$, we must have

$\mathsf{triggered}(C, A) \neq \emptyset$. In an analogous fashion one shows that $T_2 \subseteq \mathsf{trans}(S_1)$ is equivalent to $\mathsf{triggered}(D, A) = \emptyset$, using the admissibility of T_2 for $S_1 \parallel D$. This proves that Prop. (A.13) directly follows from Prop. (A.11).

We are now left with the task of verifying that T_2 is admissible for $S \parallel D$. We know that T_2 is admissible for $S_1 \parallel D$ and that T_1 is admissible for $S \parallel C$. Also, $S_1 = T_1 \cap \mathsf{trans}(S) = T_2 \cap \mathsf{trans}(S)$ and $\mathsf{act}(T_1) = \mathsf{act}(T_2) = A$. We calculate as follows:

$$\begin{aligned}
&\mathsf{enabled}(S \parallel D, \emptyset, T_2) \cap \mathsf{trans}(S) \\
&\quad = \mathsf{consistent}(S \parallel D, T_2) \cap \mathsf{triggered}(S \parallel D, A) \cap \mathsf{trans}(S) \\
&\quad = \mathsf{consistent}(S, T_2 \cap \mathsf{trans}(S)) \cap \mathsf{triggered}(S \parallel D, A) \cap \mathsf{trans}(S) \\
&\quad = \mathsf{consistent}(S, T_1 \cap \mathsf{trans}(S)) \cap \mathsf{triggered}(S \parallel C, A) \cap \mathsf{trans}(S) \\
&\quad = \mathsf{consistent}(S \parallel C, T_1) \cap \mathsf{triggered}(S \parallel C, A) \cap \mathsf{trans}(S) \\
&\quad = \mathsf{enabled}(S \parallel C, \emptyset, T_1) \cap \mathsf{trans}(S) \\
&\quad = T_1 \cap \mathsf{trans}(S) \\
&\quad = T_2 \cap \mathsf{trans}(S)\,.
\end{aligned} \tag{A.14}$$

The next to last equation follows from the admissibility of T_1 for $S \parallel C$. Moreover, we have

$$\begin{aligned}
&\mathsf{enabled}(S \parallel D, \emptyset, T_2) \cap \mathsf{trans}(D) \\
&\quad = \mathsf{consistent}(S \parallel D, T_2) \cap \mathsf{triggered}(S \parallel D, A) \cap \mathsf{trans}(D) \\
&\quad = \mathsf{consistent}(D, T_2) \cap \mathsf{triggered}(D, A) \cap \mathsf{trans}(D) \\
&\quad = \mathsf{consistent}(S_1 \parallel D, T_2) \cap \mathsf{triggered}(S_1 \parallel D, A) \cap \mathsf{trans}(D) \\
&\quad = \mathsf{enabled}(S_1 \parallel D, \emptyset, T_2) \cap \mathsf{trans}(D) \\
&\quad = T_2 \cap \mathsf{trans}(D)\,.
\end{aligned} \tag{A.15}$$

The last step is due to the admissibility of T_2 for $S_1 \parallel D$. Since $T_2 \subseteq \mathsf{trans}(S \parallel D) = \mathsf{trans}(S) \cup \mathsf{trans}(D)$, Props. (A.14) and (A.15) imply $T_2 = \mathsf{enabled}(S \parallel D, \emptyset, T_2)$. Finally, the inseparability of T_2 for $S \parallel D$ follows from the fact that $\mathsf{enabled}(S_1 \parallel D, \emptyset, T') \subseteq \mathsf{enabled}(S \parallel D, \emptyset, T')$, for all $T' \subseteq T_2$, and from the inseparability of T_2 for $S_1 \parallel D$.

This completes the first part of the proof of Lemma 4.10, namely that $C \simeq_f^+ D$ implies $C \simeq_f D$.

Now we tackle the other direction "$\Longrightarrow$" of the proposition under consideration, i.e., we prove that $C \simeq_f D$ entails $C \simeq_f^+ D$. Let us assume $C \simeq_f D$. Thus, C and D have the same responses in all C_f–contexts. In particular, then, they have the same responses in all parallel C_f–contexts, i.e., $(C \parallel P) \Downarrow_E A$ if and only if $(D \parallel P) \Downarrow_E A$, for all $P \in \mathsf{PC}_f$ and $E, A \subseteq_{\mathrm{fin}} \Pi$. This is because $\Phi[x] =_{\mathrm{df}} x \parallel P$ is simply a special context in C_f, by virtue of the closure properties. To obtain $C \simeq_f^+ D$, however, we must also verify, for all responses A, that A is active for C if and only if A is active for D. We prove this property by contradiction. Suppose that, for some $P \in \mathsf{PC}$ and $E, A \subseteq_{\mathrm{fin}} \Pi$, we have $(C \parallel P) \Downarrow_E A$ and $(D \parallel P) \Downarrow_E A$, but $\rho(C, A) \neq \rho(D, A)$. We may assume w.l.o.g. that $E = \emptyset$, as E can always be accounted for in P, and that $\mathsf{triggered}(C, A) = \emptyset$ but $\mathsf{triggered}(D, A) \neq \emptyset$. Hence, no transition of C is triggered in A, but some transitions of D are. We are going to exhibit a context $\Phi[x] \in \mathsf{C}_f$ such that $\Phi[D] \Downarrow A$ but $\Phi[C] \not\Downarrow A$. Let $e \in \Pi$ be some fresh event that does not already occur in either C or D. Consider the context $\Phi[x] =_{\mathrm{df}} (t_1 + x) \parallel t_2$, where t_1 is the transition $\cdot/e$ and where t_2 is the transition $\cdot/A$. By Assumption (i) about fragment C_f, we conclude $\Phi[x] \in \mathsf{C}_f$. We claim that (1) $\Phi[D] \Downarrow A$ and (2) $\Phi[C] \not\Downarrow A$. As for (1), we argue as follows. The given response $(D \parallel P) \Downarrow A$ implies that there exists a set T of transitions which are admissible for $D \parallel P$, such that $A = \mathsf{act}(T)$. We claim that $T_1 =_{\mathrm{df}} (T \cap \mathsf{trans}(D)) \cup \{t_2\}$ is admissible for $D \parallel t_2$. First of all,

$$\begin{aligned} T_1 &= (T \cap \mathsf{trans}(D)) \cup \{t_2\} \\ &= (\mathsf{consistent}(D \parallel P, T) \cap \mathsf{triggered}(D \parallel P, A) \cap \mathsf{trans}(D)) \cup \{t_2\} \\ &= (\mathsf{consistent}(D, T) \cap \mathsf{triggered}(D, A)) \cup \{t_2\} \\ &= \mathsf{consistent}(D \parallel t_2, T_1) \cap \mathsf{triggered}(D \parallel t_2, A) \\ &= \mathsf{consistent}(D \parallel t_2, T_1) \cap \mathsf{triggered}(D \parallel t_2, \mathsf{act}(T_1)) \\ &= \mathsf{enabled}(D \parallel t_2, \emptyset, T_1)\,. \end{aligned}$$

We wish to show that T_1 is also inseparable for $D \parallel t_2$. To this end, assume $T' \subset T_1$. We must prove that there exists a transition $t \in T_1 \setminus T'$ that is triggered by the events in $\mathsf{act}(T')$. Of course, t_2 is always triggered, so if $t_2 \in T_1 \setminus T'$, then we are done. Assume $t_2 \in T'$. This implies $A \subseteq \mathsf{act}(T')$, which means by construction of T_1 that $\mathsf{act}(T')$ triggers all transitions in T_1. Hence, we may choose any transition $t \in T_1 \setminus T'$ to witness the inseparability of T_1. Thus, we have shown that T_1 is admissible for $D \parallel t_2$. We also have by our assumptions that $\mathsf{triggered}(D, A) \neq \emptyset$, i.e., at least one transition of D is enabled by A, so that $T_1 \cap \mathsf{trans}(D) \neq \emptyset$. We may now apply Lemma 4.9 (distributivity) to conclude that T_1 is admissible for $(t_1 + D) \parallel t_2 = \Phi[D]$. As $A = \mathsf{act}(T_1)$, this yields $\Phi[D] \Downarrow A$. Surely, $e \notin A$ since the transitions in T_1 does not mention event e at all.

It remains to be seen that $\Phi[C] \not\Downarrow A$. We establish this statement by showing that event e must be contained in all responses of $\Phi[C]$. In essence, we prove that the only response of $\Phi[C]$ is $A \cup \{e\}$. Let T be an admissible set of transitions for $\Phi[C] = (t_1 + C) \parallel t_2$. We first observe that T cannot include any transition from C. Otherwise, T would have to be admissible for $C \parallel t_2$ by Lemma 4.9. Clearly, $t_2 \in T$, since t_2 is unconditionally enabled and consistent with all transitions of $C \parallel t_2$. But then $\{t_2\} \subset T$ and, due to the inseparability of T, there would have to exist some transition $t \in T \setminus \{t_2\}$ that is triggered by $\mathsf{act}(\{t_2\}) = A$. This transition t would have to come from configuration C. But this is impossible since $\mathsf{triggered}(C, A) = \emptyset$ by our initial assumption, i.e., configuration C does not contain any transition enabled by A. Hence, $T \cap \mathsf{trans}(C) = \emptyset$. Lemma 4.9 then implies that T must be admissible for $t_1 \parallel t_2$. Since both transitions t_1 and t_2 are unconditionally enabled and consistent with each other, we have $T = \{t_1, t_2\}$. Thus, any response $\mathsf{act}(T)$, for every admissible set T for $\Phi[C]$, must be identical to $A \cup \{e\}$. □

Appendix B. Proof of Lemma 4.4 (Expansion). For notational convenience, we introduce the abbreviations $C =_{\mathrm{df}} (P + Q) \parallel R$ and $D =_{\mathrm{df}} (\mathsf{watch}(P, Q) \parallel P \parallel R) + (\mathsf{watch}(Q, P) \parallel Q \parallel R)$. We tacitly assume that the transitions in both copies of R in the expansion D are *named apart*. To indicate the two copies of R we use the notations R^l and R^r for the left and right occurrences, respectively. By Prop. 4.1, $C \simeq D$ if and only if for all parallel configurations $S \in \mathsf{PC}$ and $A \subseteq_{\mathrm{fin}} \Pi$:

1. $(C \parallel S) \Downarrow A$ implies $(D \parallel S) \Downarrow A$ and $\rho(C, A) = \rho(D, A)$.
2. $(D \parallel S) \Downarrow A$ implies $(C \parallel S) \Downarrow A$ and $\rho(C, A) = \rho(D, A)$.

It is easy to see that $\rho(C, A) = \rho(D, A)$ whenever A is a response of both $C \parallel S$ and $D \parallel S$. The only possible situation where some transition in one of C and D is triggered but none in the other, would be when A enabled one of the watchdogs in D, as these transitions are not contained in C. But this cannot be the case since then response A would contain event $\bot$. In fact, no transitions of any watchdog can ever be enabled in a response A. Thus, condition $\rho(C, A) = \rho(D, A)$ holds in Statements (1) and (2).

For Statement (1), we assume that $(C \parallel S) \Downarrow A$, i.e., $((P + Q) \parallel R \parallel S) \Downarrow A$. Further, we let $T \subseteq \mathsf{trans}((P + Q) \parallel R \parallel S)$ be a corresponding set of admissible transitions with $\mathsf{act}(T) = A$. By Lemma 4.9, we have the following three cases to consider:

$$T \text{ is admissible for } (P \parallel R \parallel S)\,,\text{ and } T \cap \mathsf{trans}(P) \neq \emptyset\,. \tag{B.1}$$

$$T \text{ is admissible for } (Q \parallel R \parallel S)\,,\text{ and } T \cap \mathsf{trans}(Q) \neq \emptyset\,. \tag{B.2}$$

$$T \text{ is admissible for both } P \parallel R \parallel S \text{ and } Q \parallel R \parallel S\,,\text{ and } T \subseteq \mathsf{trans}(R \parallel S)\,. \tag{B.3}$$

In Case (B.1), $\mathsf{triggered}(P, A) \neq \emptyset$, so the watchdog $\mathsf{watch}(P, Q)$ is switched off, i.e., all transitions are disabled. Hence, T is admissible for $\mathsf{watch}(P, Q) \parallel P \parallel R^l \parallel S$. Please recall the watchdog property (cf. Prop. 4.3), namely that T is admissible for $\mathsf{watch}(P, Q) \parallel P \parallel R^l \parallel S$ if and only if T is admissible for $P \parallel R^l \parallel S$ and $\mathsf{triggered}(P, A) \neq \emptyset$ or $\mathsf{triggered}(Q, A) = \emptyset$, where $A = \mathsf{act}(T)$. But T is admissible for $\mathsf{watch}(P, Q) \parallel P \parallel R^l \parallel S$ and $T \cap \mathsf{trans}(P) \neq \emptyset$, i.e., $T \cap \mathsf{trans}(\mathsf{watch}(P, Q) \parallel P \parallel R^l) \neq \emptyset$, which implies $(D \parallel S) \Downarrow A$ by Lemma 4.9. This proves Statement (1) in Case (B.1). Case (B.2) is completely symmetric. Finally, consider Case (B.3), in which no transitions of P or Q can be enabled, so both watchdogs $\mathsf{watch}(P, Q)$ and $\mathsf{watch}(Q, P)$ are switched off. Then, T must be admissible for both $\mathsf{watch}(P, Q) \parallel P \parallel R^l \parallel S$ and $\mathsf{watch}(Q, P) \parallel Q \parallel R^r \parallel S$, when invoking the watchdog property (cf. Prop. 4.3). If $T \subseteq \mathsf{trans}(S)$, then T is admissible for $D \parallel S$ by Lemma 4.9. This shows $(D \parallel S) \Downarrow A$. If $T \cap \mathsf{trans}(R) \neq \emptyset$, then T is admissible for $\mathsf{watch}(P, Q) \parallel P \parallel R^l \parallel S$. From Lemma 4.9 again, we infer that T must be admissible for $D \parallel S$, so $(D \parallel S) \Downarrow A$, as desired.

Now we show Statement (2) starting from $(D \parallel S) \Downarrow A$. Let T with $\mathsf{act}(T) = A$ be an admissible set of transitions for $((\mathsf{watch}(P, Q) \parallel P \parallel R^l) + (\mathsf{watch}(Q, P) \parallel Q \parallel R^r)) \parallel S$. Again, we use Lemma 4.9 to distinguish the following three possible situations:

$$T \text{ is admissible for } \mathsf{watch}(P, Q) \parallel P \parallel R^l \parallel S\,,\text{ and } T \cap \mathsf{trans}(\mathsf{watch}(P, Q) \parallel P \parallel R^l) \neq \emptyset\,. \tag{B.4}$$

$$T \text{ is admissible for } \mathsf{watch}(Q, P) \parallel Q \parallel R^r \parallel S\,,\text{ and } T \cap \mathsf{trans}(\mathsf{watch}(Q, P) \parallel Q \parallel R^r) \neq \emptyset\,. \tag{B.5}$$

$$T \text{ is admissible for } \mathsf{watch}(P, Q) \parallel P \parallel R^l \parallel S \text{ and } \mathsf{watch}(Q, P) \parallel Q \parallel R^r \parallel S\,,\text{ and } T \subseteq \mathsf{trans}(S)\,. \tag{B.6}$$

In Case (B.4), if T is admissible for $\mathsf{watch}(P, Q) \parallel P \parallel R^l \parallel S$, then no transition of watchdog $\mathsf{watch}(P, Q)$ can be enabled; otherwise, $A = \mathsf{act}(T)$ would contain event $\bot$. Thus, $T \subseteq \mathsf{trans}(P \parallel R^l \parallel S)$, whence $T \cap \mathsf{trans}(P \parallel R^l) = T \cap \mathsf{trans}(\mathsf{watch}(P, Q) \parallel P \parallel R^l) \neq \emptyset$, and T is admissible for $P \parallel R^l \parallel S$. Consequently, T is admissible for $P \parallel R \parallel S$. As $\mathsf{watch}(P, Q)$ is disabled by A, we know that P must be enabled or Q be disabled. In the first case, $T \cap \mathsf{trans}(P) \neq \emptyset$, so by application of Lemma 4.9 we conclude that T is admissible for $(P + Q) \parallel R \parallel S$, which proves $(C \parallel S) \Downarrow A$. If, however, $T \cap \mathsf{trans}(P) = \emptyset$, then we also have $T \cap \mathsf{trans}(Q) = \emptyset$ since then Q is disabled by A. This means $T \subseteq \mathsf{trans}(R^l \parallel S)$ and T must be admissible for both $P \parallel R \parallel S$ and $Q \parallel R \parallel S$. Lemma 4.9 yields that T is admissible for $C \parallel S$, whence $(C \parallel S) \Downarrow A$. Case (B.5) is symmetric and, therefore, omitted. Case (B.6) remains to be checked. In this situation, the response stems from S alone, as no transition in $(\mathsf{watch}(P, Q) \parallel P \parallel R^l) + (\mathsf{watch}(Q, P) \parallel Q \parallel R^r)$ is enabled. As a consequence, no transition in $P + Q$ is enabled, either. Since T is admissible for both $\mathsf{watch}(P, Q) \parallel P \parallel R^l \parallel S$ and $\mathsf{watch}(Q, P) \parallel Q \parallel R^r \parallel S$, by the properties of watchdogs (cf. Prop. 4.3), T is admissible for $P \parallel R^l \parallel S$ and $Q \parallel R^r \parallel S$ at the same time. Moreover, since T does not contain any transitions from R^r and R^l, it must then be admissible for both $P \parallel R \parallel S$ and $Q \parallel R \parallel S$. Finally, $T \subseteq \mathsf{trans}(S)$ implies $T \subseteq \mathsf{trans}(R \parallel S)$, so that Lemma 4.9 may be used to show that T is admissible for $C \parallel S$. Hence, $(C \parallel S) \Downarrow A$. This completes the proof of Lemma 4.4.

Appendix C. Alternative Approach: Encoding the Choice Operator. Another way for obtaining a fully–abstract semantics for Statecharts, based on our results for the parallel fragment presented in Sec. 3, is to eliminate the choice operator by syntactic encoding using the parallel operator. The advantage of such a method over the one employed in Sec. 4 is that we can use the simpler semantics of the parallel fragment of our configuration algebra, namely behaviors, as opposed to the more involved contexts and enabling information. However, the alternative method is, again, not purely semantic as it involves a syntactic transformation process.

The key observation for the work in this section is that the choice operator $+$ may be eliminated in terms of the parallel operator $\|$ by using transition names as special events. Intuitively, the event represented by a transition name $t \in \mathcal{T}$, which we refer to as *transition event*, indicates that "*transition t has fired.*" Technically, we let $\Pi^+ =_{\text{df}} \Pi \cup \mathcal{T}$ denote the extended set of events. The idea behind our encoding is to implement the constraint of the operational semantics, which governs the handling of $+$, explicitly in terms of transition events. More precisely, for every configuration C and set $T \subseteq \mathcal{T}$ of transition names, we define a parallel configuration C^+_T that is equivalent to C under the assumption that no transition from T is executed together with a transition in C, i.e., parameter T in translation C^+_T represents a set of transitions that must be orthogonal to all transitions in C. In our encoding we achieve this by adding T as negative trigger events to all transitions in C. At the same time, we add to every transition t in C its transition name t as a new event to its action, so that whenever t is fired this fact is signaled to the environment. Transition event t can be used by the environment to block those transitions that are not orthogonal, or inconsistent, to t. As an example, consider configuration

$$((t_1 : a/b + t_2 : b/c) \,\|\, t_3 : c/d)^+_\emptyset \;=_{\text{df}}\; (t_1 : \bar{t}_2, a/b, t_1) \,\|\, (t_2 : \bar{t}_1, b/c, t_2) \,\|\, t_3 : c/d\,.$$

The mutual exclusion of transitions t_1 and t_2 is now generated by the event signaling scheme regarding transition events t_1 and t_2. However, in the encoding, actions will no longer be uniquely determined by transition names. Hence, the actions generated by a given set T of transitions now depend on configuration C. To account for this, we replace $\mathsf{act}(T)$ by the notation $\mathsf{generated}(C, T)$ in the sequel.

We now formalize our translation. Let $T \subseteq_{\text{fin}} \mathcal{T}$ be a finite set of transition names and C be an arbitrary configuration. We define the encoding C^+_T of C relative to T inductively along the structure of C.

$$\begin{aligned} (t : P, \overline{N}/A)^+_T &=_{\text{df}} t : P, \overline{N \cup T}/A \cup \{t\} \\ (C_1 \,\|\, C_2)^+_T &=_{\text{df}} (C_1)^+_T \,\|\, (C_2)^+_T \\ (C_1 + C_2)^+_T &=_{\text{df}} (C_1)^+_{T \cup \mathsf{trans}(C_2)} \,\|\, (C_2)^+_{T \cup \mathsf{trans}(C_1)} \end{aligned}$$

For notational convenience, we often write C^+ instead of $C^+_\emptyset$. Observe that C and C^+ have exactly the same transition names, i.e., $\mathsf{trans}(C) = \mathsf{trans}(C^+)$. The difference between the two configurations is that transitions in C^+ have additional negative triggers and action events. More precisely, each transition $t : P, \overline{N}/A$ in C corresponds to transition $t : P, \overline{N \cup N'}/A, t$ in C^+, where $N' =_{\text{df}} \mathsf{trans}(C) \setminus \mathsf{consistent}(C, \{t\})$. In other words, the additional negative triggers are the names of all transitions which are in conflict with t, and the extra action is the transition name t. Hence, $\mathsf{generated}(C^+, T) = \mathsf{generated}(C, T) \cup T$. The equivalence between C and C^+ is highlighted by the following lemma which implies that a finite set T of transitions is E–admissible for C if and only if it is E–admissible for C^+.

LEMMA C.1. *Let C be a configuration and E a set of events, which do not contain any transition event. Then, $\mathsf{enabled}(C, E, T) = \mathsf{enabled}(C^+, E, T)$ holds, for every set $T \subseteq \mathsf{trans}(C)$ of transitions.*

Proof. Recall that $\mathsf{trans}(C) = \mathsf{trans}(C^+)$. Let $T \subseteq \mathsf{trans}(C)$ be chosen arbitrarily. The equation $\mathsf{enabled}(C, E, T) = \mathsf{enabled}(C^+, E, T)$ is equivalent to

$$\begin{aligned}&\mathsf{consistent}(C, T) \cap \mathsf{triggered}(C, E \cup \mathsf{generated}(C, T)) = \\ &\qquad \mathsf{consistent}(C^+, T) \cap \mathsf{triggered}(C^+, E \cup \mathsf{generated}(C^+, T))\,.\end{aligned}$$

Let $t \in \mathsf{consistent}(C, T) \cap \mathsf{triggered}(C, E \cup \mathsf{generated}(C, T))$. As $C^+ \in \mathsf{PC}$, we have $\mathsf{consistent}(C^+, T) = \mathsf{trans}(C^+)$, whence $t \in \mathsf{consistent}(C^+, T)$. It remains to show $t \in \mathsf{triggered}(C^+, E \cup \mathsf{generated}(C^+, T))$. We know that, if t is of form $t : P, \overline{N}/A$ in C, then it must look like $t : P, \overline{N \cup N'}/A, t$ in C^+, where $N' =_{\mathrm{df}} \mathsf{trans}(C) \setminus \mathsf{consistent}(C, \{t\})$. Since $t \in \mathsf{triggered}(C, E \cup \mathsf{generated}(C, T))$ and $\mathsf{generated}(C^+, T) = \mathsf{generated}(C, T) \cup T$, we just need to show that $N \cap T = \emptyset$ and $N' \cap (\mathsf{generated}(C, T) \cup T) = \emptyset$. The former follows from the fact that C does not use transition events, and also $N' \cap \mathsf{generated}(C, T) = \emptyset$ holds for the same reason. The missing piece is $N' \cap T = \emptyset$ which can be established as follows. By assumption, $t \in \mathsf{consistent}(C, T)$ and, thus, $T \subseteq \mathsf{consistent}(C, \{t\})$, by the property of consistency. But as $N' = \mathsf{trans}(C) \setminus \mathsf{consistent}(C, \{t\})$, the desired result is immediate.

For the other direction, assume $t \in \mathsf{consistent}(C^+, T) \cap \mathsf{triggered}(C^+, E \cup \mathsf{generated}(C^+, T))$, which is equivalent to $t \in \mathsf{triggered}(C^+, E \cup \mathsf{generated}(C^+, T))$ because of $\mathsf{consistent}(C^+, T) = \mathsf{trans}(C^+)$. Suppose t in C^+ has the form $t : P, \overline{N \cup N'}/A, t$, where $N' = \mathsf{trans}(C) \setminus \mathsf{consistent}(C, \{t\})$. Since t is triggered in C^+ by $E \cup \mathsf{generated}(C^+, T) = E \cup \mathsf{generated}(C, T) \cup T$, we must have

$$P \subseteq E \cup \mathsf{generated}(C, T) \cup T \text{ and} \tag{C.1}$$

$$(N \cup N') \cap (E \cup \mathsf{generated}(C, T) \cup T) = \emptyset\,. \tag{C.2}$$

As P cannot contain any transition name, Prop. (C.1) implies $P \subseteq E \cup \mathsf{generated}(C, T)$. This yields the first half of the argument that t is triggered in C by $E \cup \mathsf{generated}(C, T)$. Recall that t in C is $t : P, \overline{N}/A$. The second half, thus, is to show that $N \cap (E \cup \mathsf{generated}(C, T)) = \emptyset$. But this is an immediate consequence of Prop. (C.2), i.e., $t \in \mathsf{triggered}(C, E \cup \mathsf{generated}(C, T))$, too. It remains to be seen why $t \in \mathsf{consistent}(C, T)$. Here we can use $N' = \mathsf{trans}(C) \setminus \mathsf{consistent}(C, \{t\})$ in conjunction with Prop. (C.2). This property implies that N' and T are disjoint, which, because of $T \subseteq \mathsf{trans}(C)$, means $T \subseteq \mathsf{consistent}(C, \{t\})$. But this is the same as stating $t \in \mathsf{consistent}(C, T)$. Thus, we have shown $t \in \mathsf{consistent}(C, T) \cap \mathsf{triggered}(C, E \cup \mathsf{generated}(C, T))$, which completes our proof. □

A direct consequence of Lemma C.1 is that our encoding preserves the step semantics of configurations, up to transition names.

PROPOSITION C.2. *For all configurations C, event sets $E, A \subseteq_{\mathit{fin}} \Pi$, and contexts $\Phi[x]$, such that C and $\Phi[x]$ do not contain transition events:*

1. $\Phi[C] \Downarrow_E A$ *implies* $\exists T \subseteq \mathsf{trans}(C).\ \Phi[C^+] \Downarrow_E (A \cup T)$.
2. $\Phi[C^+] \Downarrow_E A$ *implies* $\exists T \subseteq \mathsf{trans}(C).\ \Phi[C] \Downarrow_E (A \setminus T)$.

Proof. We first prove the special case $\Phi[C] = C$ where the context is trivial but C may be arbitrary, i.e., C possibly contains transition names as events. We have to establish the following two properties:

1. $C \Downarrow_E A$ implies $\exists T \subseteq \mathsf{trans}(C).\ C^+ \Downarrow_E (A \cup T)$ and
2. $C^+ \Downarrow_E A$ implies $\exists T \subseteq \mathsf{trans}(C).\ C \Downarrow_E (A \setminus T)$.

To prove Case (1), suppose $C \Downarrow_E A$. Then, there exists an E–admissible set $T \subseteq \mathsf{trans}(C)$ of transitions from C for which $A = E \cup \mathsf{generated}(C, T)$. Since, by Lemma C.1, $\mathsf{enabled}(C, E, T) = \mathsf{enabled}(C^+, E, T)$, for all $T \subseteq \mathsf{trans}(C) = \mathsf{trans}(C^+)$, transition set T must also be an E–admissible set for C^+. We define $B =_{\mathrm{df}} E \cup \mathsf{generated}(C^+, T)$, whence $C^+ \Downarrow_E B$. Since $\mathsf{generated}(C^+, T) = \mathsf{generated}(C, T) \cup T$, we obtain $B = A \cup T$, as required. Next, consider Case (2) and assume $C^+ \Downarrow_E A$. Thus, there exists a set T of E–admissible transitions for C^+ such that $A = E \cup \mathsf{generated}(C^+, T)$. Again, by Lemma C.1, T must be E–admissible for C, too. Defining $B =_{\mathrm{df}} E \cup \mathsf{generated}(C, T)$ we have $C \Downarrow_E B$. Since $\mathsf{generated}(C^+, T) = \mathsf{generated}(C, T) \cup T$ and $\mathsf{generated}(C, T) \cap T = \emptyset$, we conclude $B = A \setminus T$. This implies $C \Downarrow_E (A \setminus T)$, which proves Case (2).

Finally, let us consider the general case. Its proof depends on the fact that for all contexts $\Phi[x]$ and configurations C:

$$(\Phi[C^+])^+ = (\Phi[C])^+, \tag{C.3}$$

which can be shown without difficulty by a separate induction on the structure of $\Phi[x]$. We also assume that $\Phi[C]$ does not contain any transition names as events. Suppose $\Phi[C] \Downarrow_E A$. Then, Case (2) implies that there exists a transition set $T \subseteq \mathsf{trans}(\Phi[C])$ satisfying $(\Phi[C])^+ \Downarrow_E ((A \cup T))$. Now, Prop. (C.3) implies $(\Phi[C^+])^+ \Downarrow_E (A \cup T)$. From this, by Prop. (2), we obtain $T' \subseteq \mathsf{trans}(\Phi[C^+]) = \mathsf{trans}(\Phi[C])$ such that $\Phi[C^+] \Downarrow_E ((A \cup T) \setminus T')$. Since $A \cap T' = \emptyset$, we obtain $(A \cup T) \setminus T' = A \cup (T \setminus T')$. This was to be shown.

For the other direction, let $\Phi[C^+] \Downarrow_E A$. As $\Phi[\cdot]$ does not have any transition names as events, all transition names in A must come from C, i.e., $A \cap \mathcal{T} \subseteq \mathsf{trans}(C)$. We employ Prop. (2) to conclude $(\Phi[C^+])^+ \Downarrow_E (A \cup T)$, for some transition set $T \subseteq \mathsf{trans}(\Phi[C^+]) = \mathsf{trans}(\Phi[C])$. Further, Prop. (C.3) implies $(\Phi[C])^+ \Downarrow_E (A \cup T)$, whence by Prop. (2), there exists $T' \subseteq \mathsf{trans}(\Phi[C])$ such that $\Phi[C] \Downarrow_E ((A \cup T) \setminus T')$. Since by assumption $\Phi[C]$ does not contain any transition names as events, we must have $T' \supseteq T$. Moreover, as A is the response of $\Phi[C^+]$, the only transition names in A are those from $\mathsf{trans}(C)$, in accordance with our assumption about the context. Hence, there exists a transition set $T'' \subseteq \mathsf{trans}(C)$ such that $A \setminus T' = A \setminus T''$. This implies $(A \cup T) \setminus T' = (A \setminus T') \cup (T \setminus T') = A \setminus T'' \cup \emptyset = A \setminus T''$, as desired. □

Let us assume that C is a configuration that does not use transition names as events. Then Prop. C.2 implies that C and C^+ have exactly the same step responses, if we ignore all transition names in the responses of the encoding. In fact, the difference between C and C^+ is that the responses of the latter also record all transitions from C that have fired to produce the given response. Finally, observe that Prop. C.2 actually states that C and C^+ have, up to transition names, the same responses in all context, whence our encoding is compositional.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY*(Leave blank)*

2. REPORT DATE
July 2000

3. REPORT TYPE AND DATES COVERED
Contractor Report

4. TITLE AND SUBTITLE
The intuitionism behind Statecharts steps

5. FUNDING NUMBERS
C NAS1-97046
WU 505-90-52-01

6. AUTHOR(S)
Gerald Lüttgen and Michael Mendler

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Institute for Computer Applications in Science and Engineering
Mail Stop 132C, NASA Langley Research Center
Hampton, VA 23681-2199

8. PERFORMING ORGANIZATION REPORT NUMBER
ICASE Report No. 2000-28

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
National Aeronautics and Space Administration
Langley Research Center
Hampton, VA 23681-2199

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
NASA/CR-2000-210302
ICASE Report No. 2000-28

11. SUPPLEMENTARY NOTES
Langley Technical Monitor: Dennis M. Bushnell
Final Report
To appear in the 27th International Colloquium on Automata, Languages, and Programming (ICALP 2000).

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Unclassified–Unlimited
Subject Category 60, 61
Distribution: Nonstandard
Availability: NASA-CASI (301) 621-0390

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

The semantics of Statecharts macro steps, as introduced by Pnueli and Shalev, lacks compositionality. This report first analyzes the compositionality problem and traces it back to the invalidity of the Law of the Excluded Middle. It then characterizes the semantics via a particular class of linear, intuitionistic Kripke models, namely stabilization sequences. This yields, for the first time in the literature, a simple fully-abstract semantics which interprets Pnueli and Shalev's concept of failure naturally. The results not only give insight into the semantic subtleties of Statecharts, but also provide a basis for an implementation, for developing algebraic theories for macro steps, and for comparing different Statecharts variants.

14. SUBJECT TERMS
Statecharts, compositionality, full abstraction, intuitionistic Kripke semantics

15. NUMBER OF PAGES
45

16. PRICE CODE
A03

17. SECURITY CLASSIFICATION OF REPORT
Unclassified

18. SECURITY CLASSIFICATION OF THIS PAGE
Unclassified

19. SECURITY CLASSIFICATION OF ABSTRACT

20. LIMITATION OF ABSTRACT